



**DISEÑO DE UN SISTEMA DE CONTROL DE PH Y TEMPERATURA PARA LOS
BIORREACTORES DE LABORATORIO EMPLEADOS EN LA PRODUCCIÓN DE
ETANOL A PARTIR DE MATERIAS PRIMAS AMILÁCEAS**

JOHN ALEXANDER PENAGOS CALDAS

**UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA MECATRÓNICA
SANTIAGO DE CALI
2010**

**DISEÑO DE UN SISTEMA DE CONTROL DE PH Y TEMPERATURA PARA LOS
BIORREACTORES DE LABORATORIO EMPLEADOS EN LA PRODUCCIÓN DE
ETANOL A PARTIR DE MATERIAS PRIMAS AMILÁCEAS**

JOHN ALEXANDER PENAGOS CALDAS

MODALIDAD: PASANTÍA INSTITUCIONAL

**DIRECTOR:
CARLOS HÉCTOR CRUZ VERGARA
INGENIERO ELECTRICISTA**

**UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA MECATRÓNICA
SANTIAGO DE CALI
2010**

Nota de aceptación

Firma del presidente del Jurado

Firma del jurado

Firma del jurado

Santiago de Cali, ____ de _____ de 2010

DEDICATORIA

Antes que nada quiero agradecer a mi madre quién ha sido la persona que más me ha apoyado durante toda mi vida de manera incondicional afectiva y económicamente, es a ella a quién quiero dedicar este trabajo en primera instancia. También hago una mención especial a mi novia, quién sin duda alguna se ha convertido en un motivo importante de superación personal y a mi abuela materna que también ha estado presente durante todo mi proceso formativo. Igualmente agradezco a mi padre por el apoyo económico que me ha brindado durante todo mi proceso formativo.

Quiero dedicar éste trabajo a todas las personas que de algún modo han contribuido a mi formación integral tales como docentes, familiares, amigos, colegas, etc. Pues gracias a sus enseñanzas es que me encuentro preparado para ser un profesional integral.

JOHN ALEXANDER PENAGOS CALDAS

AGRADECIMIENTOS

- A mi madre por haberme apoyado desde el principio de manera incondicional.
- A Sonia Gallego asistente de investigación de CLAYUCA CIAT, por su ayuda incondicional y por todos sus aportes en la realización del proyecto.
- A Víctor Moreno y Diego Guerrero por su amistad incondicional y apoyo en los momentos más importantes del desarrollo del proyecto.
- A Carlos Héctor Cruz por sus aportes para el desarrollo del proyecto.
- A todo el personal de CLAYUCA CIAT que me prestó apoyo en la realización del proyecto.

TABLA DE CONTENIDO

	Pág.
RESUMEN.....	15
ABSTRACT	16
INTRODUCCIÓN.....	17
1 PLANTEAMIENTO DEL PROBLEMA	18
2 JUSTIFICACIÓN	19
3 OBJETIVOS	20
3.1 OBJETIVO GENERAL.....	20
3.2 OBJETIVOS ESPECÍFICOS.....	20
4 MARCO TEÓRICO	21
4.1 LOS BIORREACTORES	21
4.2 LOS MICROCONTROLADORES	21
4.2.1 Diferencia entre microprocesador y microcontrolador	21
4.2.2 Núcleo de un microcontrolador	23
4.2.3 Arquitecturas de computadora.....	23
4.2.4 Procesador en detalle.....	24
4.2.5 Memoria	26
4.2.6 Interrupciones.....	27
4.2.7 Periféricos	28

4.3	MOTORES DE PASO.....	30
4.3.1	Motor de reluctancia variable.....	31
4.3.2	Motor de imán permanente.....	32
4.3.3	Motor híbrido.	38
4.4	BOMAS PERISTÁLTICAS.....	38
4.5	MEDICIÓN DE pH.....	44
4.5.1	Tipos de Electrodo de Medición.....	46
4.5.2	Efectos de la temperatura sobre la medición de pH.....	47
4.6	CONTROL INTELIGENTE.....	48
4.6.1	Lógica difusa	49
4.6.2	Controladores difusos.....	52
4.7	REGULACIÓN POR ÁNGULO DE FASE	58
4.7.1	Tensión eficaz V_{RMS}	58
4.7.2	Relación entre la tensión eficaz V_{RMS} y el ángulo de retardo.....	58
4.7.3	Relación entre la tensión eficaz V_{RMS} y el ángulo de retardo.....	58
5	CAPITULO 1. Desarrollo de las etapas del diseño concurrente para el sistema de control de temperatura y pH del biorreactor de laboratorio	61
5.1	Desarrollo conceptual.....	61
5.1.1	Identificación de las necesidades.	61
5.1.2	Establecer especificaciones preliminares	63
5.1.3	Generar conceptos.....	64
5.1.4	Seleccionar conceptos.....	65

5.1.5	Establecer especificaciones finales	77
5.2	Diseño a nivel de sistemas	78
5.2.1	Estudio y selección de la arquitectura.....	78
5.3	Diseño detallado.....	80
5.3.1	Diseño para manufactura (DPM).	80
5.3.2	Diseño para mantenimiento (DPMT).....	82
5.3.3	Diseño Físico del dispositivo automatizado	82
5.3.4	Diseño del circuito electrónico	85
5.3.5	Diseño del Software.	93
6	CONCLUSIONES.....	114
7	BIBLIOGRAFÍA.....	115
8	ANEXOS	117

LISTA DE FIGURAS

	Pág.
Figura 1. Sistema básico microprocesado.....	22
Figura 2. Sistema básico microcontrolado.....	22
Figura 3. Microcontrolador con memoria EPROM.	27
Figura 4. Vista de un motor de reluctancia variable.....	31
Figura 5. Funcionamiento de un motor de imán permanente.	32
Figura 6. Distribución del bobinado de un motor unipolar.....	33
Figura 7. Distribución del bobinado de un motor bipolar.....	37
Figura 8. Puente H.....	38
Figura 9. Disposición típica de la bomba peristáltica.	39
Figura 10. Diseño de una bomba peristáltica comercial.	40
Figura 11. Funcionamiento de las bombas peristálticas.	40
Figura 12. Bombas peristálticas para el manejo de líquidos.....	42
Figura 13. Bombas peristálticas industriales.	42
Figura 14. Bombas peristálticas para el bombeo de tinta.	43
Figura 15. Montaje experimental para realizar medidas potenciométricas.	45
Figura 16. Electrodo de Referencia.....	46
Figura 17. Electrodos Indicadores.....	47
Figura 18. Efecto de la temperatura en la pendiente del sensor.....	48
Figura 19. Error cometido en la medida de pH sin compensación de la temperatura.	48
Figura 20. Conjunto binario vs conjunto difuso.....	49
Figura 21. Función triangular.	50
Figura 22. Función Γ	50
Figura 23. Función sigmoideal	51
Figura 24. Función gaussiana.....	51

Figura 25. Función trapezoidal.....	51
Figura 26. Conceptos de lógica difusa.	52
Figura 27. Esquema de un sistema retroalimentado de control clásico.....	53
Figura 28. Esquema de controlador difuso.....	53
Figura 29. Ejemplo de una situación en un instante t.	53
Figura 30. Funcionamiento de un sistema difuso tipo Takagi Sujeno.	55
Figura 31. Representación de la salida de un controlador difuso con funciones Singleton.	55
Figura 32. Esquema para la implementación de un controlador difuso.	56
Figura 33. Sintonización por medio de conjuntos difusos.....	57
Figura 34. Efecto regulatorio del control por ángulo de fase.	59
Figura 35. Relación entre la tensión pico y la tensión eficaz para ángulos de conducción comúnmente utilizados.	60
Figura 36. Despliegue de la función de calidad (QFD).	63
Figura 37. Representación de las entradas y salidas del sistema de control del biorreactor.	64
Figura 38. Descomposición funcional del sistema del control.....	65
Figura 39. PIC18f4520.....	69
Figura 40. PIC16f877.....	69
Figura 41. Electrodo EPH-M11-Pt100-HT.	71
Figura 42. Electrodo AP301-2120134.	71
Figura 43. Trasmisor HT-DMM4000-pH.	72
Figura 44. Trasmisor PHTX-271-1P.	73
Figura 45. Bomba peristáltica SR25-S300.	74
Figura 46. Bomba peristáltica M045 D.C.....	74
Figura 47. Válvula solenoide 2W040.....	77
Figura 48. Válvula solenoide EV210A.	77

Figura 49. Relación entre elementos funcionales y elementos físicos del sistema de control.....	79
Figura 50. Recipiente del biorreactor.	83
Figura 51. Intercambiador de calor.....	83
Figura 52. Ensamble del intercambiador la resistencia y el recipiente.....	83
Figura 53. Dispositivo de control.	84
Figura 54. Biorreactor con el sistema de automatización.	85
Figura 55. Fuente de 5V DC.	86
Figura 56. Sumador no inversor.....	88
Figura 57. Señal de salida del sumador no inversor.....	88
Figura 58. Esquema de conexión al microcontrolador del bloque de detección de cruce por cero.	89
Figura 59. Etapa de potencia para la resistencia.....	90
Figura 60. Etapa de potencia para la válvula del intercambiador de calor.....	91
Figura 61. Esquema de la implementación del L298.....	92
Figura 62. Esquema de conexión entre el PIC18f4520 y el L298.	92
Figura 63. Esquema de la interfaz hombre máquina.	93
Figura 64. Configuración del módulo comparador correspondiente al valor 01H cargado en el registro CMCON.....	94
Figura 65. Diagrama esquemático de la simulación del detector de cruce por cero realizado con el PIC18f84520.....	95
Figura 66. Comportamiento del detector de cruce por cero realizado con PIC18f4520.	95
Figura 67. Diagrama de flujo de la tarea Temp.	96
Figura 68. Universos de discurso del error y la derivada de la temperatura para el controlador de temperatura.....	98

Figura 69. Representación de los conjuntos difusos según el formato empleado en los arreglos.	98
Figura 70. Diagrama de flujo del método Calcular_Pertenencia().	100
Figura 71. Universo de discurso de la salida del controlador.....	102
Figura 72. Recorrido de los vectores en el método defuzz().	103
Figura 73. Diagrama de flujo del método defuzz().....	104
Figura 74. Diagrama esquemático de la simulación realizada para probar el método Angulo_Disparo().....	107
Figura 75. Configuración del generador de funciones.	107
Figura 76. Resultados de la simulación de prueba del método Angulo_Disparo()....	108
Figura 77. Diagrama esquemático de la simulación de la tarea Temp.	109
Figura 78. Comportamiento del horno simulado en Proteus 7.6 con controlador difuso.	109
Figura 79. Diagrama de flujo de la tarea pH.....	111

LISTA DE TABLAS

	Pág.
Tabla 1. Secuencia tipo Wave Drive.....	34
Tabla 2. Secuencia Normal.	34
Tabla 3. Secuencia de medio paso.....	35
Tabla 4. Secuencia para motores bipolares.....	38
Tabla 5. Identificación de las necesidades.	61
Tabla 6. Métricas de las necesidades.....	62
Tabla 7. Criterios y nivel de importancia para la selección del concepto “Recibir y transformar energía”.	68
Tabla 8. Selección del concepto “Recibir y transformar energía”.	68
Tabla 9. Criterios y nivel de importancia para la selección del concepto “Dispositivo de control”.	70
Tabla 10. Selección del concepto “Dispositivo de Control”.	70
Tabla 11. Criterios y nivel de importancia para la selección del concepto “Sensor pH”.	71
Tabla 12. Selección del concepto “Sensor pH”.	72
Tabla 13. Criterios y nivel de importancia para la selección del concepto “Instrumentación de la señal de pH”.	73
Tabla 14. Selección del concepto “Instrumentación de la señal de pH”.	73
Tabla 15. Criterios y nivel de importancia para la selección del concepto “Instrumentación de la señal de pH”.	75
Tabla 16. Selección del concepto “Actuadores de pH”.	75
Tabla 17. Criterios y nivel de importancia para la selección del concepto “Etapa de potencia de los actuadores de pH”.	76

Tabla 18. Selección del concepto “Etapa de potencia de los actuadores de pH”.....	76
Tabla 19. Lista general de componentes.	80
Tabla 20. Lista general de componentes.	81
Tabla 21. Costo total del dispositivo.	82
Tabla 22. Acciones de mantenimiento.....	82
Tabla 23. Rango de valores del ADC según el tipo de dispositivo y la directiva #device ADC=.....	97
Tabla 24. Formato de los arreglos que contienen a los conjuntos difusos del error y la derivada de la temperatura.	98
Tabla 25. Formato del arreglo que contiene los valores de pertenencia a los conjuntos difusos.	99
Tabla 29. Formato del vector SalidaResistencia.....	102
Tabla 27. Secuencia de medio paso para motores bipolares.	112

LISTA DE ANEXOS

	Pág.
Anexo 1. ESQUEMA DEL CIRCUITO ELECTRÓNICO.....	118
Anexo 2. CÓDIGO PRUEBA ÁNGULO DISPARO	119
Anexo 3. CÓDIGO PARA LA PRUEBA DE LA TAREA TEMP	122
Anexo 4. TABLA DE DATOS PARA LA TAREA ÁNGULO DISPARO	127
Anexo 5. PLANOS DEL RECIPIENTE DEL BIORREACTOR.....	130
Anexo 6. PLANOS DE LA CARCAZA DEL DISPOSITIVO DE CONTROL.....	131
Anexo 7. CÓDIGO IMPLEMENTADO EN EL SISTEMA DE CONTROL	132

RESUMEN

En el presente documento se presenta el desarrollo del método de ingeniería concurrente aplicado en el desarrollo de un sistema de control de pH y temperatura para los biorreactores de laboratorio con los que cuenta CLAYUCA – CIAT, realizando en primera instancia, un estudio de cada uno de los conceptos necesarios para la elaboración del sistema de control. Una vez determinados en su totalidad los elementos que componen el sistema se planteó la estrategia de control más acertada, y se programó la interfaz hombre – máquina de manera sencilla para darle al operario la oportunidad de manipular el sistema de manera práctica.

ABSTRACT

This paper presents the development of concurrent engineering method applied in the development of a control system of pH and temperature for the laboratory bioreactors which account CLAYUCA - CIAT, performing in the first instance, a study of each concept necessities for the developing of the control system. Once in full certain elements of the system raised the most successful control strategy, and the HMI interface was programmed to give to the operator the opportunity to manipulate the system in practice.

INTRODUCCIÓN

En este documento se muestra el desarrollo del proceso de diseño del sistema de control para los biorreactores de laboratorio con los que cuenta CLAYUCA – CIAT, además se explican conceptos necesarios para el entendimiento de las estrategias de control implementadas en el sistema mencionado. El desarrollo del diseño se encuentra descrito utilizando la metodología de diseño concurrente, y se explica en detalle cada una de las etapas comprometidas en el desarrollo como lo son: despliegue de la función de calidad (DFQ), diseño para la manufactura (DFM), diseño para ensamble (DFA), y diseño para el mantenimiento (DFMT).

1 PLANTEAMIENTO DEL PROBLEMA

Los biorreactores de laboratorio contruidos por CLAYUCA no cuentan con un sistema de control de temperatura ni de pH, lo cual es indispensable para aumentar la precisión y exactitud de los resultados que se obtienen en las pruebas realizadas en el equipo.

2 JUSTIFICACIÓN

Con la implementación del sistema de control pH y temperatura en el biorreactor, CLAYUCA va a obtener lo siguiente:

- Aumento en la confiabilidad de los resultados que arrojen las pruebas realizadas en el equipo.
- Disminución en el consumo de energía debido a ensayos que no mantuvieron la muestra en las condiciones ideales.

3 OBJETIVOS

3.1 OBJETIVO GENERAL

Diseñar la automatización de los biorreactores de laboratorio, equipos experimentales para la producción de etanol a partir de yuca y batata.

3.2 OBJETIVOS ESPECÍFICOS

- Analizar si las variables a controlar son dependientes o independientes.
- Diseñar controladores de pH y temperatura para los biorreactores.
- Realizar un modelado CAD del dispositivo automatizado
- Realizar un presupuesto para el sistema de control.
- Simular el comportamiento de los actuadores para probar los controladores.

4 MARCO TEÓRICO

4.1 LOS BIORREACTORES

Un biorreactor es un recipiente o sistema que mantiene un ambiente biológicamente activo. En algunos casos, un biorreactor es un recipiente en el que se lleva a cabo un proceso químico que involucra organismos o sustancias bioquímicamente activas derivadas de dichos organismos.¹

En términos generales, un biorreactor busca mantener ciertas condiciones ambientales propicias, tales como:

- pH.
- Temperatura.
- Concentración de Oxígeno.

Estas entre muchas otras condiciones que pueden ser implantadas para un producto específico.

4.2 LOS MICROCONTROLADORES

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de una computadora completa en un solo circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria. Sus líneas de entrada/salida soportan la conexión de los sensores y actuadores del dispositivo a controlar.

Son diseñados para disminuir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la CPU, la cantidad de memoria y los periféricos incluidos dependen de la aplicación.

Los microcontroladores se encuentran en casi cualquier dispositivo electrónico, como automóviles, lavadoras, hornos microondas, teléfonos, etc. Los microcontroladores representan la inmensa mayoría de los chips de computadoras vendidos, sobre un 50% son controladores "simples" y el restante corresponde a DSPs más especializados.

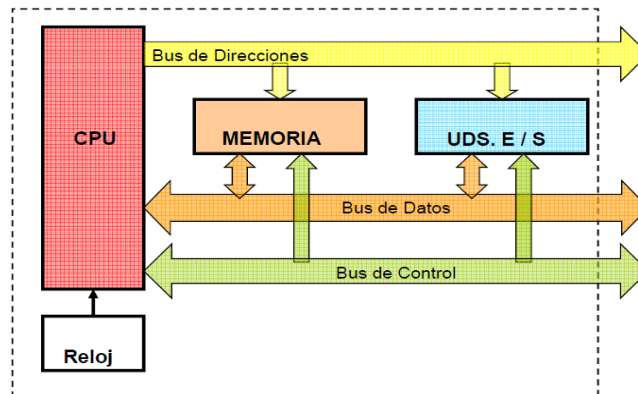
4.2.1 Diferencia entre microprocesador y microcontrolador. El microprocesador es un circuito integrado que contiene la Unidad Central de Procesamiento (CPU), también llamada procesador de un computador. La CPU está formada por la Unidad de Control que interpreta las instrucciones y el camino de datos que las ejecuta. Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control para permitir conectarle con la memoria y los módulos de E/S y configurar un computador complementado por varios circuitos integrados como se observa en la Figura 1. Un microprocesador es un sistema abierto porque su

¹ Wikipedia. *La enciclopedia libre*. [En línea] [Citado el: 06 de 01 de 2010.] <http://es.wikipedia.org/wiki/Biorreactor>.

configuración es variable de acuerdo con la aplicación a la que se destine.

En un sistema abierto basado en un microprocesador, la disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación. Si sólo se dispusiese de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones.

Figura 1. Sistema básico microprocesado.

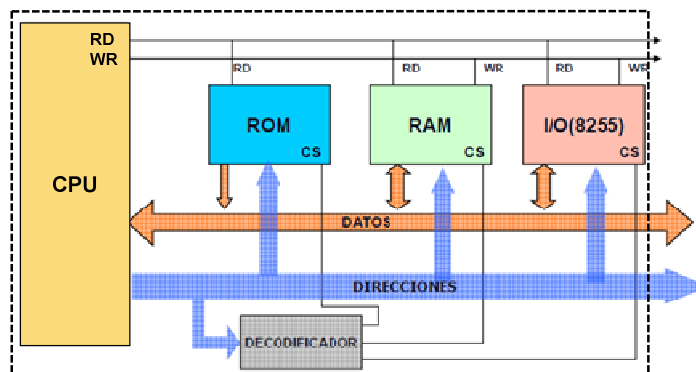


Fuente: Pérez Villa, José Fernando. *Introducción a los Microprocesadores y Microcontroladores*. [Documento PDF] Santiago de Cali, Colombia. 2007.

Un microcontrolador es más fácil de convertir en una computadora en funcionamiento con un mínimo de chips externos de apoyo. La idea es que el chip se coloque en el dispositivo enganchado a la fuente de energía y de la información que necesite. Un microprocesador tradicional no permite hacer esto, ya que espera que todas estas tareas sean manejadas por otros chips. Hay que agregarle los módulos de entrada/salida (puertos) y memoria para almacenamiento de información.

El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

Figura 2. Sistema básico microcontrolado.



Fuente: Pérez Villa, José Fernando. *Introducción a los Microprocesadores y Microcontroladores*. [Documento PDF] Santiago de Cali, Colombia. 2007.

En la Figura 2, vemos al microcontrolador metido dentro de un encapsulado de circuito integrado, con su procesador (CPU), buses, memoria, periféricos y puertos de entrada/salida. Fuera del encapsulado se ubican otros circuitos para completar periféricos internos y dispositivos que pueden conectarse a los pines de entrada/salida. También se conectarán a los pines del encapsulado la alimentación, masa, circuito de completamiento del oscilador y otros circuitos necesarios para que el microcontrolador pueda trabajar.

En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

4.2.2 Núcleo de un microcontrolador. Aún cuando el microcontrolador es una computadora embebida dentro de un circuito integrado, se compone de un núcleo y un conjunto de circuitos adicionales. Dentro del núcleo se encuentran el procesador y la memoria, todo ello estructurado de forma tal que conforme una arquitectura de computadora.

4.2.3 Arquitecturas de computadora. Básicamente existen dos arquitecturas de computadoras, y por supuesto, están presentes en el mundo de los microcontroladores: Von Neumann y Harvard. Ambas se diferencian en la forma de conexión de la memoria al procesador y en los buses que cada una necesita.

4.2.3.1 Arquitectura Von Neumann. La arquitectura desarrollada por Jon Von Neumann se caracteriza por tener una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. La CPU se conecta a través de un sistema de buses (direcciones, datos y control). Esta arquitectura es limitada cuando se demanda rapidez.

4.2.3.2 Arquitectura Harvard. Fue desarrollado en Harvard, por Howard Aiken, esta arquitectura se caracteriza por tener dos memorias independientes: una que contiene sólo instrucciones y otra, que contiene sólo datos. Ambas disponen de sus respectivos sistemas de buses para el acceso y es posible realizar operaciones de acceso simultáneamente en ambas memorias.

Las principales ventajas de esta arquitectura son:

- El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.
- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad de operación.

La desventaja está en que consume muchas líneas de E/S del procesador; por lo que en sistemas donde el procesador está ubicado en su propio encapsulado, solo se utiliza en supercomputadoras. Sin embargo, en los microcontroladores y otros sistemas embebidos, donde usualmente la memoria de datos y programas comparten el mismo encapsulado que el procesador, este inconveniente deja de ser un problema serio y es por ello que encontramos la arquitectura Harvard en la mayoría de los microcontroladores.

4.2.4 Procesador en detalle. A continuación se describen las partes más importantes que constituyen un microcontrolador.

4.2.4.1 Registros. Son un espacio de memoria muy reducido pero necesario para cualquier microprocesador, de aquí se toman los datos para varias operaciones que debe realizar el resto de los circuitos del procesador. Los registros sirven para almacenar los resultados de la ejecución de instrucciones, cargar datos desde la memoria externa o almacenarlos en ella.

4.2.4.2 Unidad de control. Esta unidad es una de las más importantes en el procesador porque en ella recae la lógica necesaria para la decodificación y ejecución de las instrucciones, el control de los registros, la ALU, los buses y cualquier elemento extra que se desee integrar en el procesador.

La unidad de control es uno de los elementos fundamentales que determinan las prestaciones del procesador ya que su tipo y estructura determinan parámetros tales como el tipo de conjunto de instrucciones, velocidad de ejecución, tiempo del ciclo de máquina, tipo de buses que puede tener el sistema, manejo de interrupciones y un buen número de cosas más que en cualquier procesador van a parar a este bloque.

Por supuesto, las unidades de control son el elemento más complejo de un procesador y normalmente están divididas en unidades más pequeñas trabajando en conjunto. La unidad de control agrupa componentes tales como la unidad de decodificación, unidad de ejecución, controladores de memoria caché, controladores de buses, controladores de interrupción, pipelines, entre otros elementos, dependiendo siempre del tipo de procesador.

4.2.4.3 Unidad aritmético-lógica. Como los procesadores son circuitos que hacen básicamente operaciones lógicas y matemáticas, se le dedica a este proceso una unidad completa, con cierta independencia. Aquí es donde se realizan las sumas, restas y operaciones lógicas típicas del álgebra de Boole.

Su impacto en las prestaciones del procesador es también importante porque dependiendo de su potencia, tareas más o menos complejas pueden hacerse en tiempos muy cortos, como por ejemplo, los cálculos en coma flotante.

4.2.4.4 Buses. Son el medio de comunicación que utilizan los diferentes componentes del procesador para intercambiar información entre sí, eventualmente los buses o una parte de ellos estarán reflejados en los pines del encapsulado del procesador.

En el caso de los microcontroladores, no es común que los buses estén reflejados en el encapsulado del circuito ya que estos se destinan básicamente a las E/S de propósito general y periféricos del sistema.

Existen tres tipos de buses:

- Dirección: Se utiliza para seleccionar al dispositivo con el cual se quiere trabajar o en el caso de las memorias, seleccionar el dato que se desea leer o escribir.
- Datos: Se utiliza para enviar y recibir datos.
- Control: Se utiliza para gestionar los distintos procesos de escritura lectura y controlar la operación de los dispositivos del sistema.

4.2.4.5 Conjunto de instrucciones. El conjunto de instrucciones define las operaciones básicas que puede realizar el procesador, que conjugadas y organizadas forman lo que conocemos como software. El conjunto de instrucciones vienen siendo como las letras del alfabeto, el elemento básico del lenguaje, que organizadas adecuadamente permiten escribir palabras, oraciones, entre otras.

Existen dos tipos básicos de repertorios de instrucciones, que determinan la arquitectura del procesador: CISC y RISC.

El repertorio CISC, del inglés Complex Instruction Set Computer, se denominan; Computadora de Conjunto de Instrucciones Complejo. Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y por permitir realizar operaciones complejas entre operandos situados en la memoria o en los registros internos. Este tipo de repertorio dificulta el paralelismo entre instrucciones, por lo que en la actualidad, la mayoría de los sistemas CISC de alto rendimiento convierten las instrucciones complejas en varias instrucciones simples del tipo RISC llamadas generalmente microinstrucciones.

Por otro lado, las instrucciones RISC, del inglés Reduced Instruction Set Computer, se denominan; Computadora con Conjunto de Instrucciones Reducido, se centra en la obtención de procesadores con las siguientes características fundamentales:

- Instrucciones de tamaño fijo.
- Pocas instrucciones.
- Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.
- Número relativamente elevado de registros de propósito general.

Una de las características más destacables de este tipo de procesadores es que posibilitan el paralelismo en la ejecución, y reducen los accesos a memoria. Es por eso que los procesadores más modernos tradicionalmente basados en arquitecturas CISC implementan mecanismos de traducción de instrucciones CISC a RISC para aprovechar las ventajas de este tipo de procesadores.

4.2.5 Memoria. Anteriormente se vio que la memoria en los microcontroladores debe estar ubicada dentro del mismo encapsulado, esto es así la mayoría de las veces porque la idea fundamental es mantener el grosor de los circuitos del sistema dentro de un solo integrado.

En los microcontroladores la memoria no es abundante, aquí no encontrará Gigabytes de memoria como en las computadoras personales. Típicamente la memoria de programas no excederá de 16 Kilobytes de memoria no volátil para instrucciones y la memoria RAM ni siquiera llegará a exceder los 5 Kilobytes.

La memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además los registros de trabajo del procesador y los de configuración y trabajo de los distintos periféricos del microcontrolador. Es por ello que en la mayoría de los casos, aunque se tenga un espacio de direcciones de un tamaño determinado, la cantidad de memoria RAM de que dispone el programador para almacenar sus datos es menor que la que puede direccionar el procesador.

El tipo de memoria utilizada en las memorias RAM de los microcontroladores es SRAM, lo que evita tener que implementar sistemas de refrescamiento como en el caso de las computadoras personales, que utilizan gran cantidad de memoria, típicamente alguna tecnología DRAM. A pesar de que la memoria SRAM es más costosa que la DRAM, es el tipo adecuado para los microcontroladores porque éstos poseen pequeñas cantidades de memoria RAM.

En el caso de la memoria de programas se utilizan diferentes tecnologías. El uso de una u otra depende de las características de la aplicación a desarrollar, a continuación se describen las cinco tecnologías existentes, que mayor utilización tienen o han tenido, hasta el momento:

- ROM de máscara: En este caso no se “graba” el programa en memoria sino que el microcontrolador se fabrica con el programa, es un proceso similar al de producción de los CD comerciales mediante masterización. El costo inicial de producir un circuito de este tipo es alto, porque el diseño y producción de la máscara es un proceso costoso, sin embargo, cuando se necesitan varios miles o incluso cientos de miles de microcontroladores para una aplicación determinada, como por ejemplo, algún electrodoméstico, el costo inicial de producción de la máscara y el de fabricación del circuito se distribuye entre todos los circuitos de la serie y el costo final de ésta es bastante menor que el de sus semejantes con otro tipo de memoria.
- TP (One Time Programmable): Este tipo de memoria también es conocida como PROM o simplemente ROM. Los microcontroladores con memoria OTP se pueden programar una sola vez y son usados en sistemas donde el programa no requiere futuras actualizaciones y para series relativamente pequeñas donde la variante de máscara sea muy costosa, también para sistemas que requieren serialización de datos, almacenados como constantes en la memoria de programas.
- EPROM (Erasable Programmable Read Only Memory): Los microcontroladores con este tipo de memoria son muy fáciles de identificar porque su encapsulado

es de cerámica y llevan encima una ventanita de vidrio desde la cual puede verse la oblea de silicio del microcontrolador como se observa en la Figura 3. Se fabrican así porque la memoria EPROM es reprogramable, pero antes debe borrarse, y para ello hay que exponerla a una fuente de luz ultravioleta, el proceso de grabación es similar al empleado para las memorias OTP. Al aparecer tecnologías menos costosas y más flexibles como las memorias EEPROM y FLASH, este tipo de memoria usada en sistemas que requieren actualizaciones del programa y para los procesos de desarrollo y puesta a punto han caído en desuso.

Figura 3. Microcontrolador con memoria EPROM.



Fuente: Blogspot. [En línea] [Citado el: 11 de 02 de 2010.] http://4.bp.blogspot.com/_js6wgtUcfdQ/R4O4QP0aVRI/AAAAAAAAABqY/-Ei4TxsbBKc/s400/memoria_eprom_27C256_32K.jpg.

- **EEPROM (Electrical Erasable Programmable Read Only Memory):** Fueron el sustituto natural de las memorias EPROM, la diferencia fundamental es que pueden ser borradas eléctricamente, por lo que la ventanilla de cristal de cuarzo y los encapsulados cerámicos no son necesarios. Al disminuir los costos de los encapsulados, los microcontroladores con este tipo de memoria se hicieron más baratos y cómodos para trabajar que sus equivalentes con memoria EPROM. Otra característica destacable de este tipo de microcontrolador es que fue en ellos donde comenzaron a utilizarse los sistemas de programación en circuito o ICSP (In Circuit Serial Programming) que evitan tener que sacar el microcontrolador de la tarjeta que lo aloja para hacer actualizaciones al programa.
- **FLASH:** En el campo de las memorias reprogramables para microcontroladores, son el último avance tecnológico en uso a gran escala y han sustituido a los microcontroladores con memoria EEPROM. A las ventajas de las memorias FLASH se le adicionan su gran densidad respecto a sus predecesoras lo que permite incrementar la cantidad de memoria de programas a un costo muy bajo. Pueden además ser programadas con las mismas tensiones de alimentación del microcontrolador, el acceso en lectura y la velocidad de programación es superior, disminución de los costos de producción.

4.2.6 Interrupciones. Las interrupciones constituyen quizá el mecanismo más importante para la conexión del microcontrolador con el mundo exterior, sincronizando la ejecución de programas con acontecimientos externos, por medio de un circuito específico encargado de detectar un evento en particular.

Es de resaltar que cada microcontrolador cuenta con un conjunto de circuitos que le permiten detectar eventos externos así que es recomendable revisar la hoja de datos para saber que fuentes de interrupción posee el microcontrolador

En términos generales, un proceso de interrupción y su atención por parte del procesador, tiene la siguiente secuencia de acciones:

- En el mundo real se produce el evento para el cual queremos que el procesador ejecute un programa especial, este proceso tiene la característica de que no puede esperar mucho tiempo antes de ser atendido o no sabemos en qué momento debe ser atendido.
- El circuito encargado de detectar la ocurrencia del evento se activa, y como consecuencia, activa la entrada de interrupción del procesador.
- La unidad de control detecta que se ha producido una interrupción y “levanta” una bandera para registrar esta situación, de esta forma si las condiciones que provocaron el evento desaparecen y el circuito encargado de detectarlo desactiva la entrada de interrupción del procesador, ésta se producirá de cualquier modo, porque ha sido registrada.
- La unidad de ejecución termina con la instrucción en curso y justo antes de comenzar a ejecutar la siguiente comprueba que se ha registrado una interrupción.
- Se desencadena un proceso que permite guardar el estado actual del programa en ejecución y saltar a una dirección especial de memoria de programas, donde está la primera instrucción de la subrutina de atención a interrupción.
- Se ejecuta el código de atención a interrupción, esta es la parte “consciente” de todo el proceso porque es donde se realizan las acciones propias de la atención a la interrupción y el programador juega su papel.
- Cuando en la subrutina de atención a interrupción se ejecuta la instrucción de retorno, se desencadena el proceso de restauración del procesador al estado en que estaba antes de la atención a la interrupción.

Como podemos observar, el mecanismo de interrupción es bastante complicado, sin embargo tiene dos ventajas que obligan a su implementación: la velocidad y su capacidad de ser asincrónico. Ambas de conjunto permiten que aprovechemos al máximo las capacidades de trabajo de nuestro procesador.

Las interrupciones son tan eficaces que permiten que el procesador actúe como si estuviese haciendo varias cosas a la vez cuando en realidad se dedica a la misma rutina de siempre, ejecutar instrucciones una detrás de la otra.

4.2.7 Periféricos. Los periféricos son aquellos circuitos que podemos encontrar en el encapsulado del microcontrolador que tienen como tarea realizar una función especial tal como realizar conversión A/D, conteo de eventos externos, temporizar, etc.

4.2.7.1 Entradas y salidas de propósito general. También conocidos como puertos de E/S, generalmente agrupadas en puertos de 8 bits de longitud, permiten leer datos del exterior o escribir en ellos desde el interior del microcontrolador, el destino habitual es el trabajo con dispositivos simples como relés, LED, o cualquier otra cosa que se le ocurra al programador.

Algunos puertos de E/S tienen características especiales que le permiten manejar salidas con determinados requerimientos de corriente, o incorporan mecanismos especiales de interrupción para el procesador.

Típicamente cualquier pin de E/S puede ser considerada E/S de propósito general, pero como los microcontroladores no pueden tener infinitos pines, ni siquiera todos los pines que queramos, las E/S de propósito general comparten los pines con otros periféricos. Para usar un pin con cualquiera de las características a él asignadas debemos configurarlo mediante los registros destinados a ello.

4.2.7.2 Temporizadores y contadores. Son circuitos sincrónicos para el conteo de los pulsos que llegan a su entrada de reloj. Si la fuente de conteo es el oscilador interno del microcontrolador es común que no tengan un pin asociado, y en este caso trabajan como temporizadores. Por otra parte, cuando la fuente de conteo es externa, entonces tienen asociado un pin configurado como entrada, este es el modo contador.

Los temporizadores son uno de los periféricos más habituales en los microcontroladores y se utilizan para muchas tareas, como por ejemplo, la medición de frecuencia, implementación de relojes, para el trabajo de conjunto con otros periféricos que requieren una base estable de tiempo entre otras funcionalidades. Es frecuente que un microcontrolador típico incorpore más de un temporizador/contador e incluso algunos tienen arreglos de contadores.

4.2.7.3 Conversor analógico/digital. Es un dispositivo electrónico capaz de convertir una entrada analógica de voltaje en un valor binario, en otras palabras, este se encarga de transformar señales analógicas de entrada a digitales en la salida².

Los conversores A/D son uno de los periféricos más codiciados en el mundo de los microcontroladores y las resoluciones más frecuentes son 8 y 10 bits, aunque hay microcontroladores con conversores de 11 y 12 bits, para resoluciones mayores es preciso utilizar conversores A/D externos.

4.2.7.4 Puerto de comunicación serial. El destino común de este periférico es la comunicación con otro microcontrolador o con una PC y en la mayoría de los casos hay que agregar circuitos externos para completar la interfaz de comunicación.

La forma más común de completar el puerto serie es para comunicarlo con una PC mediante la interfaz EIA-232 (más conocida como RS-232), es por ello que muchas personas se refieren a la UART o USART como puerto serie RS-232, pero esto

² Wikipedia. *La enciclopedia libre*. [En línea] [Citado el: 15 de 02 de 2010.] http://es.wikipedia.org/wiki/Conversor_anal%C3%B3gico-digital.

constituye un error, puesto que este periférico se puede utilizar para interconectar dispositivos mediante otros estándares de comunicación.

Este periférico está presente en casi cualquier microcontrolador, normalmente en forma de UART (Universal Asynchronous Receiver Transmitter) o USART (Universal Synchronous Asynchronous Receiver Transmitter) dependiendo de si permiten o no el modo sincrónico de comunicación.

4.2.7.5 Puerto de comunicación serial sincrónica. Este tipo de periférico se utiliza para comunicar al microcontrolador con otros microcontroladores o con periféricos externos conectados a él, mediante las interfaces SPI (Serial Peripheral Interface) o I2C (Inter-Integrated Circuit).

A pesar de que es también un tipo de puerto serial, es costumbre tratarlo de forma diferenciada respecto a la UART/USART porque las interfaces SPI e I2C aparecieron mucho después que la UART/USART, su carácter es únicamente sincrónico y no están diseñadas para interconectar al sistema con otros dispositivos independientes como una PC, sino para conectar al microcontrolador dispositivos tales como memorias, pantallas LCD, conversores A/D o D/A.

4.2.7.6 Otros puertos de comunicación. En un mundo cada vez más orientado a la interconexión de dispositivos, han aparecido muchas interfaces de comunicación y los microcontroladores no se han quedado atrás para incorporarlas, es por ello que podemos encontrar algunos modelos con puertos USB (Universal Serial Bus), CAN (Controller Area Network), Ethernet, puerto paralelo entre otros.

4.2.7.7 Comparadores. Son circuitos analógicos basados en amplificadores operacionales que tienen la característica de comparar dos señales analógicas y dar como salida los niveles lógicos '0' o '1' en dependencia del resultado de la comparación.

4.2.7.8 Modulador de ancho de pulsos. Los PWM (Pulse Width Modulator) son periféricos muy útiles sobre todo para el control de motores, sin embargo hay un grupo de aplicaciones que pueden realizarse con este periférico, dentro de las cuales podemos citar: la conversión digital analógica D/A, el control regulado de luz (dimming) entre otras.

4.2.7.9 Memoria de datos no volátil. Muchos microcontroladores han incorporado este tipo de memoria como un periférico más, para el almacenamiento de datos de configuración o de los procesos que se controlan. Esta memoria es independiente de la memoria de datos tipo RAM o la memoria de programas, en la que se almacena el código del programa a ejecutar por el procesador del microcontrolador.

4.3 MOTORES DE PASO

El motor paso a paso es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa es que es capaz de avanzar una serie de grados (paso) dependiendo de sus entradas de control.

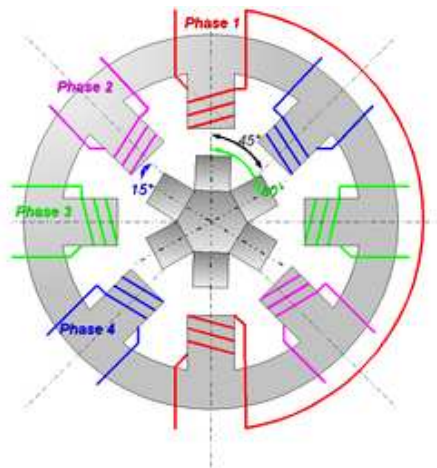
El motor paso a paso se comporta de la misma manera que un convertidor digital-analógico y puede ser gobernado por impulsos procedentes de sistemas lógicos³.

Este motor presenta las ventajas de tener alta precisión en cuanto al posicionamiento. Entre sus principales aplicaciones destacan como motor de frecuencia variable, motor de corriente continua sin escobillas, servomotores y motores controlados digitalmente.

Existen tres tipos fundamentales de motores paso a paso: el motor de reluctancia variable, el motor de imán permanente, y el motor híbrido.

4.3.1 Motor de reluctancia variable. La expresión motor de reluctancia variable hace referencia a un motor eléctrico del tipo paso a paso, cuyo funcionamiento se basa en la reluctancia⁴ variable mediante un rotor dentado en hierro dulce que tiende a alinearse con los polos bobinados del estator.⁵

Figura 4. Vista de un motor de reluctancia variable.



Fuente: Wikipedia. *La enciclopedia libre*. [En línea] [Citado el: 03 de 02 de 2010.] http://es.wikipedia.org/wiki/Motor_de_reluctancia_variable.

Este tipo de consiste en un rotor y un estator cada uno con un número diferente de dientes como se muestra en la Figura 4. Ya que el rotor no dispone de un magneto permanente el mismo gira libremente, o sea que no tiene torque de detención. A pesar de que la relación del torque a la inercia es buena, el torque dado para un tamaño de armazón dado es restringido, por lo tanto tamaños pequeños tamaños de armazones son generalmente usados y los mismos raramente varían para aplicaciones industriales⁶.

³ Wikipedia. *La enciclopedia libre*. [En línea] [Citado el: 17 de 01 de 2010.] http://es.wikipedia.org/wiki/Motor_paso_a_paso.

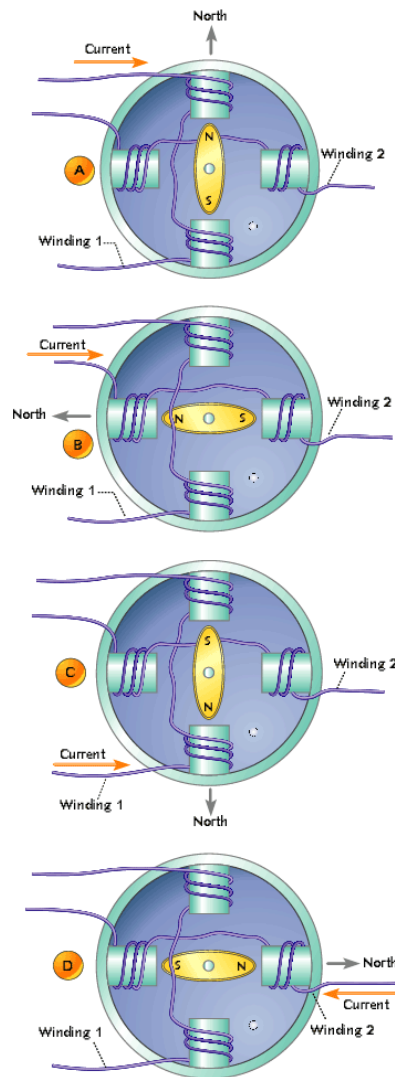
⁴ La reluctancia magnética de un material, es la resistencia que éste posee al verse influenciado por un campo magnético.

⁵ Wikipedia. *La enciclopedia libre*. [En línea] [Citado el: 03 de 02 de 2010.] http://es.wikipedia.org/wiki/Motor_de_reluctancia_variable.

⁶ **Martínez, Mario.** Universidad Autónoma de Nuevo León. [En línea] [Citado el: 17 de 01 de 2010.] <http://www.mecanica-uanl.com/pdfs/Stepper%20Motors.pdf>.

4.3.2 Motor de imán permanente. El motor de magneto permanente es quizá el motor por pasos mas ampliamente usado para aplicaciones no industriales. Básicamente, están constituidos por un rotor sobre el que van aplicados distintos imanes permanentes, y por un cierto número de bobinas excitadoras bobinadas en su estator. Así, las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) debe ser externamente manejada por un controlador, y siguiendo alguna secuencia en particular, como se muestra en la Figura 5. Los motores de imán permanente se dividen en unipolares y bipolares.

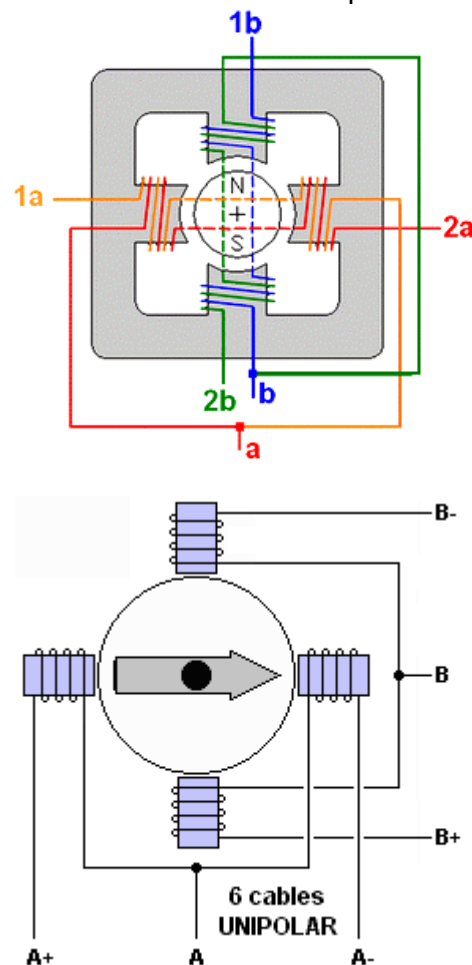
Figura 5. Funcionamiento de un motor de imán permanente.



Fuente: Foros de Electronica. [En línea] [Citado el: 03 de 01 de 2010.]
<http://www.forosdeelectronica.com/f19/introduccion-motores-paso-paso-289/>.

4.3.2.1 Motores unipolares. Los motores unipolares son relativamente fáciles de controlar, gracias a que poseen devanados duplicados. Aunque para facilitar el esquema se dibuja este devanado como una bobina con punto medio, en realidad tienen dos bobinas en cada eje del estator, que están unidas por extremos opuestos (Figura 6), de tal modo que al ser alimentada una u otra, generan cada una un campo magnético inverso al de la otra. Nunca se energizan juntas: por eso lo correcto es decir que tienen una doble bobina, en lugar de decir (como se hace habitualmente) que es una bobina con punto medio. Esta duplicación se hace para facilitar el diseño del circuito de manejo, ya que permite el uso, en la parte de potencia, de un transistor único por cada uno de los bobinados.

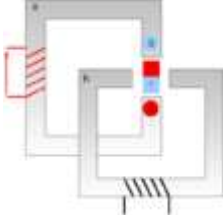
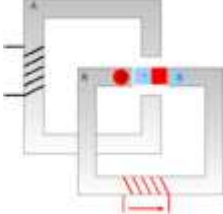
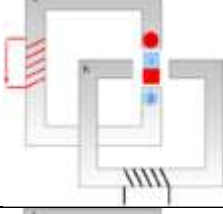
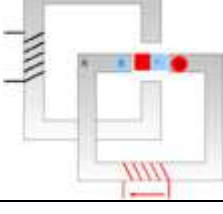
Figura 6. Distribución del bobinado de un motor unipolar.



Fuente: Carletti, Eduardo. Robots Argentina. [En línea] [Citado el: 15 de 01 de 2010.] http://robots-argentina.com.ar/MotorPP_basico.htm.

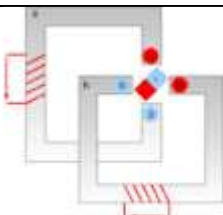
4.3.2.2 Secuencias de control para motores unipolares. A continuación se ilustran tres diferentes formas de controlar el giro de un motor unipolar, estas son:

Tabla 1. Secuencia tipo Wave Drive

No. De Paso	1A	1B	2A	2B	Imagen
Paso 1	1	0	0	0	
No. De Paso	1A	1B	2A	2B	Imagen
Paso 2	0	1	0	0	
Paso 3	0	0	1	0	
Paso 4	0	0	0	1	

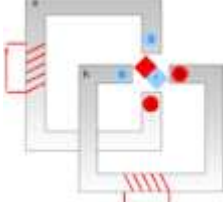
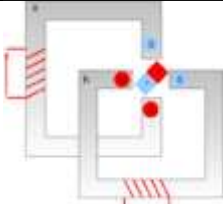
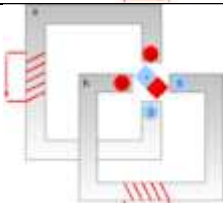
En esta secuencia se activa solo una bobina a la vez. En algunos motores esto brinda un funcionamiento más suave. La contrapartida es que al estar solo una bobina activada, el torque de paso y retención es menor⁷.

Tabla 2. Secuencia Normal.

No. De Paso	1A	1B	2A	2B	Imagen
Paso 1	1	1	0	0	

⁷ **Muquillaza, Rubén.** www.monografias.com. [En línea] [Citado el: 17 de 01 de 2010.] <http://www.monografias.com/trabajos37/motores/motores2.shtml>.

Tabla 2. (Continuación).

No. De Paso	1A	1B	2A	2B	Imagen
Paso 2	0	1	1	0	
Paso 3	0	0	1	1	
Paso 4	1	0	0	1	

Esta es la secuencia más usada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activadas, se obtiene un torque de 1.5 con respecto a la secuencia wave drive, sin embargo los movimientos del motor son muy bruscos y el consumo de corriente es aproximadamente el doble.

Tabla 3. Secuencia de medio paso.

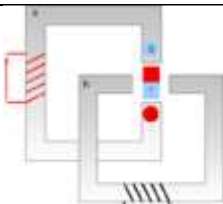
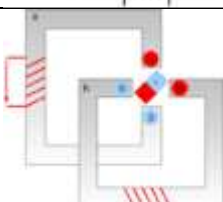
No. De Paso	1A	1B	2A	2B	Imagen
Paso 1	1	0	0	0	
Paso 2	1	1	0	0	

Tabla 3. (Continuación).

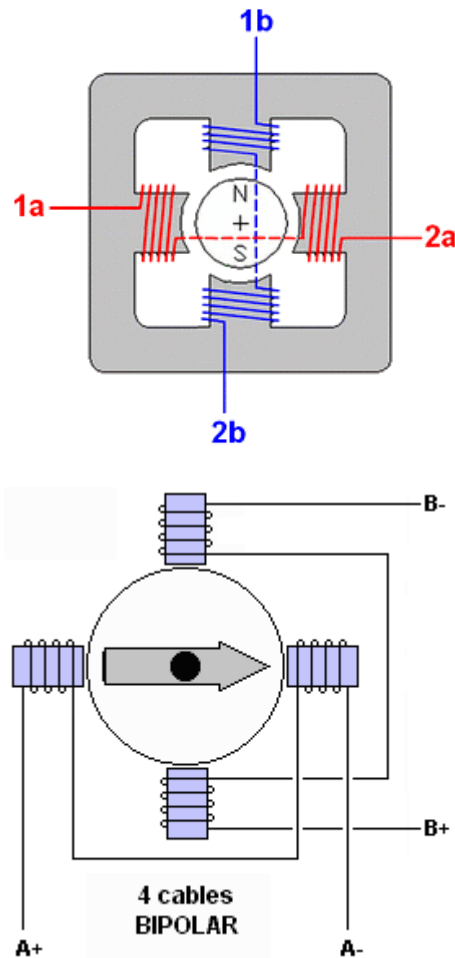
No. De Paso	1A	1B	2A	2B	Imagen
Paso 4	0	1	1	0	
Paso 5	0	0	1	0	
Paso 6	0	0	1	1	
Paso 7	0	0	0	1	
Paso 8	1	0	0	1	

La opción tres es una combinación de las dos secuencias anteriores y se denomina secuencia de medio paso. Emplear esta secuencia hace que el número de pasos por revolución sea el doble y el ángulo por paso se reduzca a la mitad.

En las tablas el número 1 representa conexión a +Vcc y el 0 representa conexión a tierra.

4.3.2.3 Motores bipolares. Estos tienen generalmente cuatro cables de salida como se observa en al Figura 7. Necesitan ciertos trucos para ser controlados, debido a que requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento.

Figura 7. Distribución del bobinado de un motor bipolar.

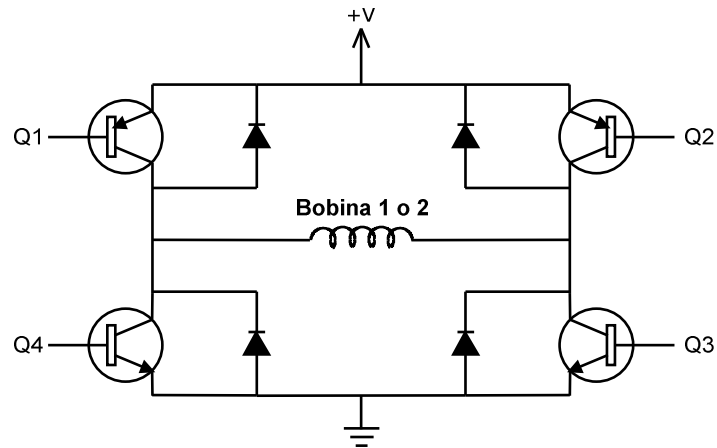


Fuente: Carletti, Eduardo. Robots Argentina. [En línea] [Citado el: 15 de 01 de 2010.] http://robots-argentina.com.ar/MotorPP_basico.htm.

Los motores bipolares requieren circuitos de control y de potencia más complejos como lo es un puente H (

Figura 8). Pero en la actualidad esto no es problema, ya que estos circuitos se suelen implementar en un integrado, que soluciona esta complejidad en un solo componente. Como mucho se deben agregar algunos componentes de potencia, como transistores y diodos para las contracorrientes, aunque esto no es necesario en motores pequeños y medianos.

Figura 8. Puente H.



4.3.2.4 Secuencias de control para motores bipolares

Tabla 4. Secuencia para motores bipolares.

No. De Paso	1A	1B	2A	2B
Paso 1	1	0	1	0
Paso 2	1	0	0	1
Paso 3	0	1	0	1
Paso 4	0	1	1	0

4.3.3 Motor híbrido. La expresión motor paso a paso híbrido refiere a un motor eléctrico del tipo paso a paso, cuyo funcionamiento se basa en la combinación de los otros dos tipos de motores paso a paso, el motor de reluctancia variable y el motor de magnetización permanente.

En este tipo de motor el rotor suele estar constituido por anillos de acero dulce dentado en un número ligeramente distinto al del estator y dichos anillos montados sobre un imán permanente dispuesto axialmente. El tipo Híbrido es probablemente el más usado de todos los motores por pasos. Originalmente desarrollado como un motor de magnetización permanente sincrónico de baja velocidad, su construcción es una combinación de los diseños del último y de reluctancia variable. El motor Híbrido consiste en un estator dentado y un rotor de tres partes (apilado simple). El rotor de apilado simple contiene dos piezas de polos separados por un magneto permanente magnetizado, con los dientes opuestos desplazados en una mitad de un salto de diente para permitir una alta resolución de pasos.

4.4 BOMAS PERISTÁLTICAS

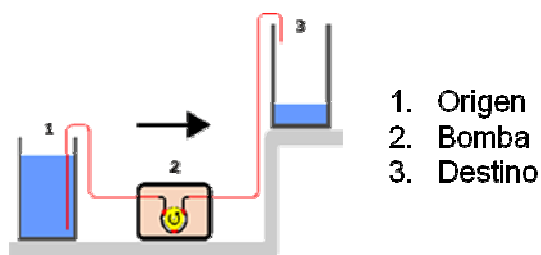
Las bombas peristálticas representan la aplicación de uno de las tecnologías más simples de bombeo, dicha creación se introdujo en el mercado a mediados de la década de 1950 aunque para ese entonces tenían muchas limitaciones debido a que las mangueras y los materiales de los tubos eran poco confiables; afortunadamente la

tecnología moderna ha proporcionado nuevos compuestos de caucho, tubería termoplástica resistente químicamente y técnicas de manufactura para obtener bombas confiables que permitan una diversa variedad de aplicaciones⁸.

Una bomba peristáltica es un tipo de bomba de desplazamiento positivo usada para bombear una variedad de fluidos. El fluido es contenido dentro de un tubo flexible empotrado dentro de una cubierta circular de la bomba (aunque se han hecho bombas peristálticas lineales). Un rotor con un número de 'rodillos', 'zapatas' o 'limpiadores' unidos a la circunferencia externa comprimen el tubo flexible. Mientras que el rotor da vuelta, la parte del tubo bajo compresión se cierra (o se ocluye) forzando, de esta manera, el fluido a ser bombeado para moverse a través del tubo. Adicionalmente, mientras el tubo se vuelve a abrir a su estado natural después del paso de la leva ('restitución'), el flujo del fluido es inducido a la bomba. Este proceso es llamado peristálsis y es usado en muchos sistemas biológicos como el aparato digestivo.⁹

Es recomendable colocar la bomba por debajo del nivel del líquido a bombear (Figura 9) porque las bombas peristálticas pueden generar una diferencia de presión mayor a la salida que la generada en la entrada de líquido¹⁰.

Figura 9. Disposición típica de la bomba peristáltica.



Fuente: Quiminet. "Ventajas, usos y aplicaciones de las bombas peristálticas". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar0/ar_vcdRsDFaddsaaddsa-ventajas-usos-y-aplicaciones-de-las-bombas-peristalticas.htm

Los rodillos están unidos con el eje del motor a través de algún mecanismo, de manera que al girar el mismo, estos presionan la manguera en forma progresiva y hacen avanzar el contenido dentro de la misma (Figura 10). En este sistema el contenido que está siendo bombeado nunca está en contacto con el mecanismo, sólo con el interior del conducto. La velocidad del bombeo puede ser tan lento como lo requiere el proceso.¹¹

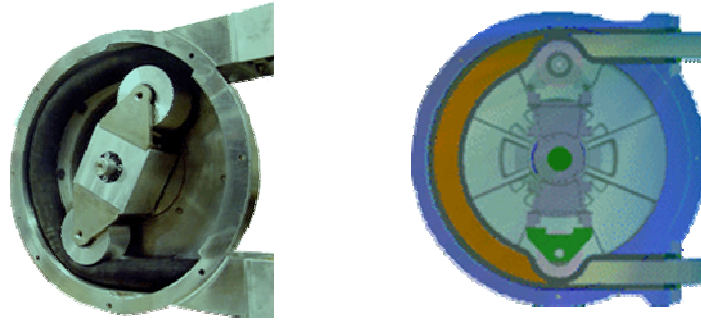
⁸ Peristaltic Pumps, Fluid Handling Solutions. [En línea] [Citado el: 11 de Enero de 2010]. <http://www.periflo.com/>

⁹ **Wikipedia.** Wikipedia, La Enciclopedia Libre. "Bomba Peristáltica". [En línea] [Citado el: 29 de Noviembre de 2009]. http://es.wikipedia.org/wiki/Bomba_perist%C3%A1ltica.

¹⁰ **Quiminet.** Quiminet. "Ventajas, usos y aplicaciones de las bombas peristálticas". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar0/ar_vcdRsDFaddsaaddsa-ventajas-usos-y-aplicaciones-de-las-bombas-peristalticas.htm

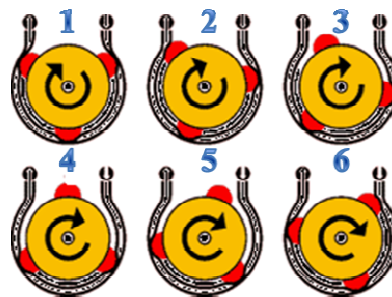
¹¹ **Quiminet.** Quiminet. "Principio de Funcionamiento de las Bombas Peristálticas". [En línea] [Citado el: 03 de Septiembre de 2009.] http://www.quiminet.com/ar9/ar_vcdarmaasdaasd-principio-del-funcionamiento-de-las-bombas-peristalticas.htm.

Figura 10. Diseño de una bomba peristáltica comercial.



Fuente: Quiminet. " *Ventajas, usos y aplicaciones de las bombas peristálticas*". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar0/ar_vcdRsDFaddsaaddsa-ventajas-usos-y-aplicaciones-de-las-bombas-peristalticas.htm

Figura 11. Funcionamiento de las bombas peristálticas.



En la Figura 11 se observa que en ningún momento de las fases 1 a 6 los rodillos dejan de presionar la manguera en al menos un punto. Esto es importante ya que si en algún momento los rodillos dejan de presionar el conducto, el líquido podría retroceder. El sentido de rotación del motor determina la dirección del flujo del contenido.

Para evitar la ruptura de la manguera interna de la bomba debido a la sobrepresión causada por el bombeo contra una válvula cerrada o al cerrarla con la bomba funcionando, o bien, cuando existan pérdidas de cargas excesivas en la descarga es importante que todas las instalaciones que cuenten con bombas peristálticas, tengan un dispositivo de parada o desvío de caudal (bypass), que alerte cuando la bomba supere la presión límite. Este dispositivo de seguridad puede ser un presostato (regulador de presión), una válvula de seguridad, un limitador de par (limitador de torque), entre otros¹².

Las bombas peristálticas son ideales para dosificar de manera controlada líquidos o pastas, los cuales son aplicados en cierto momento o de manera continua en ciertos procesos¹³ por lo que manejan una gran variedad de diferentes materiales, en las que destacan los siguientes:

¹² **Quiminet.** Quiminet. " *Las Bombas Peristálticas*". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar6/ar_vcdarmaasdzgt-las-bombas-peristalticas.htm

¹³ **Quiminet.** Quiminet. " *Las Bombas Peristálticas*". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar6/ar_vcdarmaasdzgt-las-bombas-peristalticas.htm

- **Fluidos compuestos:** Dosificación aditivos, pulpa de papel, orujo de uva (residuo sólido de la uva), piensos (alimento animal), molidos de fruta, polímeros, resinas con cuarzo.
- **Fluidos frágiles y sensibles al corte:** Emulsiones, macedonias, espumosas, tomates, puré, dosificaciones antiespuma, uvas enteras, zumos de fruta, extractos vegetales, aceites vegetales, huevos, cremas cosméticas, jabones líquidos y detergentes.
- **Fluidos contaminantes y no contaminantes:** Látex de goma, productos comestibles.
- **Fluidos abrasivos y corrosivos:** Barros y sedimentos, tartratos de cal, cementos, fibrocementos, tintas, esmaltes-pigmentos, alimentación filtros, lechada de cal, hormigón, fertilizante, residuos de establo, suspensión espesa de carbón, suspensión espesa de zeolita (aluminosilicato), hidrocarburos¹⁴.

Las bombas peristálticas se fabrican, según las necesidades específicas de su aplicación, en diferentes combinaciones de tubos flexibles, rodillos y demás elementos, pero aunque su elaboración es específica se pueden clasificar según la presión que soportan así:

- **Bombas peristálticas de alta presión:** Pueden operar con hasta 16 bar, usualmente usan zapatas. Cuentan con cubiertas con lubricante para evitar la abrasión del exterior del tubo de la bomba y ayudar a la disipación del calor. Este tipo de bombas usan tubos reforzados, a menudo llamados “mangueras”, por lo que frecuentemente son llamadas “bombas de mangueras”.
- **Bombas peristálticas de baja presión:** Generalmente tienen cubiertas secas y usan rodillos, además de tuberías no reforzadas. Este tipo de bomba, algunas veces es llamada “bomba de tubo” o “bomba de tubería”¹⁵.

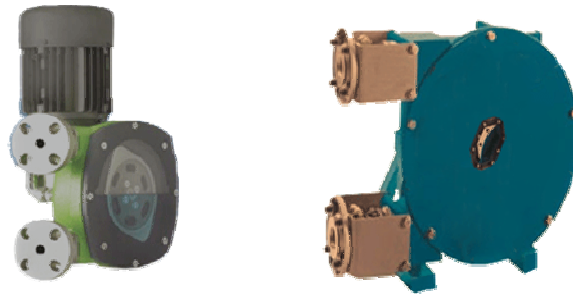
Además se clasifican según el elemento que manejan de la siguiente forma:

- **Bombas peristálticas para el manejo de líquidos,** son usadas generalmente para el manejo de líquidos viscosos y/o abrasivos, y puede abarcar los siguientes rangos de operación:
 - Viscosidades: Hasta 20,000 SSU (4'315 Cst).
 - Grado de abrasión: Lodos con hasta 50% de sólidos.
 - Capacidades: Hasta 350 GPM (477 l).
 - Presiones: Hasta 100 psi (689'475 pa).
 - Diámetros: Hasta 5" (127 mm).
 - Usos: Industria en general.

¹⁴ **Quiminet.** Quiminet. " *Ventajas, usos y aplicaciones de las bombas peristálticas*". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar0/ar_vcdRsDFadddsaadddsa-ventajas-usos-y-aplicaciones-de-las-bombas-peristalticas.htm

¹⁵ **Quiminet.** Quiminet. " *Las Bombas Peristálticas*". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar6/ar_vcdarmaasdzgt-las-bombas-peristalticas.htm

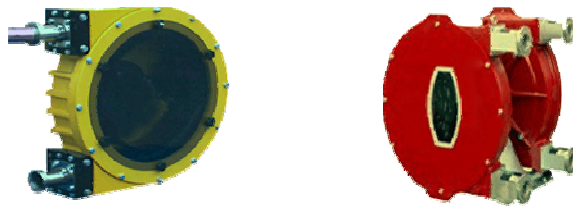
Figura 12. Bombas peristálticas para el manejo de líquidos.



Fuente: Quiminet. "Tipos principales de bombas peristálticas". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar7/ar_vcdarmaasdhgsA-tipos-principales-de-bombas-peristalticas-industriales.htm

- **Bombas peristálticas industriales**, algunas aplicaciones de las bombas peristálticas industriales se encuentran en dosificación, envasado, manejo de tintas, filtración, fabricación de cerveza, alimentación y bebidas, cosméticos, tratamiento de aguas y en la industria química.

Figura 13. Bombas peristálticas industriales.



Fuente: Quiminet. "Tipos principales de bombas peristálticas". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar7/ar_vcdarmaasdhgsA-tipos-principales-de-bombas-peristalticas-industriales.htm

- **Bombas peristálticas de la tinta**, es un dispositivo que genera un flujo presurizado usando los rodillos que rotan contra la tubería flexible especial, comprimiéndose en varios puntos en contacto con los rodillos. Son utilizadas en máquinas de impresión de escala industrial, para el control del suministro de la tinta, en imprentas, y en la industria de fabricación de tintas¹⁶.

¹⁶ **Quiminet.** Quiminet. "Tipos principales de bombas peristálticas". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar7/ar_vcdarmaasdhgsA-tipos-principales-de-bombas-peristalticas-industriales.htm

Figura 14. Bombas peristálticas para el bombeo de tinta.



Fuente: Quiminet. "Tipos principales de bombas peristálticas". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar7/ar_vcdarmaasdhgsA-tipos-principales-de-bombas-peristalticas-industriales.htm

Debido a la gran variedad de las bombas peristálticas, se pueden encontrar las siguientes ventajas, usos y aplicaciones:

- Gracias a su manguera mecanizada de alta precisión, las bombas peristálticas industriales poseen una excelente capacidad para el manejo de lodos abrasivos, ácidos corrosivos, fluidos con contenido en aire, productos de alta viscosidad, fluidos sensibles al cizallamiento y productos de alta densidad.
- El fluido se mantiene completamente separado de las partes móviles de la bomba puesto que sólo tiene contacto con el interior del tubo, por lo cual las superficies internas de la bomba son fáciles de esterilizar y limpiar, además, al no tener contacto con cierres, válvulas, diafragmas, empaquetaduras, rotores sumergidos, estatores o pistones, se evita que estos provoquen fugas, corrosión o necesidad de ser reemplazados. Son ideales para aquellas aplicaciones donde la contaminación es un punto crítico.
- Pueden funcionar en seco indefinidamente, son autocebantes, reversibles, volumétricas y, por tanto, poseen una gran precisión en la dosificación de todo tipo de fluidos. Manejan caudales de hasta 80 m³/h y presiones de hasta 16 bares.
- Son higiénicas, fáciles de instalar, limpiar y operar. Incluso en condiciones de trabajo continuo su mantenimiento es tan sencillo como un simple cambio de manguera (el único componente de la bomba sujeto a desgaste), que se realiza en pocos minutos. Sus costos de funcionamiento y el costo total de operación son extremadamente bajos en comparación con otras tecnologías de bombeo.
- Elimina el empleo de líquido refrigerante en el interior del cuerpo de la bomba, ya que el rodillo no genera fricción sobre la superficie del tubular y prolonga su vida.
- El sistema de rodillos tiene la ventaja, en presencia de productos alimenticios, que no corren ningún peligro de contaminación en el momento de rotura del tubular por la ausencia de líquido refrigerante en el interior de la bomba.
- Su carencia de válvulas, de sellos y de arandelas, y el uso de mangueras o tubos, hace que tengan un mantenimiento relativamente de bajo costo y rápido comparado a otros tipos de bombas; el cambio del elemento tubular es más limpio ya que no existe ningún líquido a evacuar y cambiar junto con el tubular.

Estas bombas ofrecen un conjunto de características que no todos los tipos de bombas tienen. Entre las que destacan los siguientes:

- Aumenta la esterilidad de la bomba y elimina la posibilidad de contaminaciones.
- Aumenta la presión de explosión brindando mayores márgenes de seguridad contra las rupturas prematuras.
- Aumenta la presión para incrementar el caudal o coeficiente de rendimiento.
- Reduce significativamente el tiempo de limpieza necesario al final de la producción de una partida comparado con las bombas de engranaje y otros tipos de bombas.
- Prácticamente elimina el riesgo de taponamiento de filtros.¹⁷

4.5 MEDICIÓN DE pH

Para determinar el pH de una solución lo que hacemos es utilizar tres elementos: dispositivo de medida de potencial, electrodo de referencia y el electrodo selectivo a iones a caracterizar.

Dispositivo de medida de potencial es el instrumento utilizado para medir el voltaje de la celda que forman el electrodo indicador y de referencia. En realidad es un voltímetro muy sensible que amplifica la débil señal eléctrica que producen los electrodos y permite lecturas de hasta 0.1 mV con gran exactitud. Los electrodos de referencia suelen ser de doble unión o de calomelano¹⁸ respecto a los cuales se miden las variaciones de potencial. Finalmente, los electrodos selectivos a iones como su nombre indica son electrodos sensibles a la presencia de un tipo de ión en la solución y permiten su cuantificación. El término “selectivo” quiere decir que son sensibles a una familia de iones de características similares, pero de entre todos ellos tienen mayor afinidad por uno en concreto.

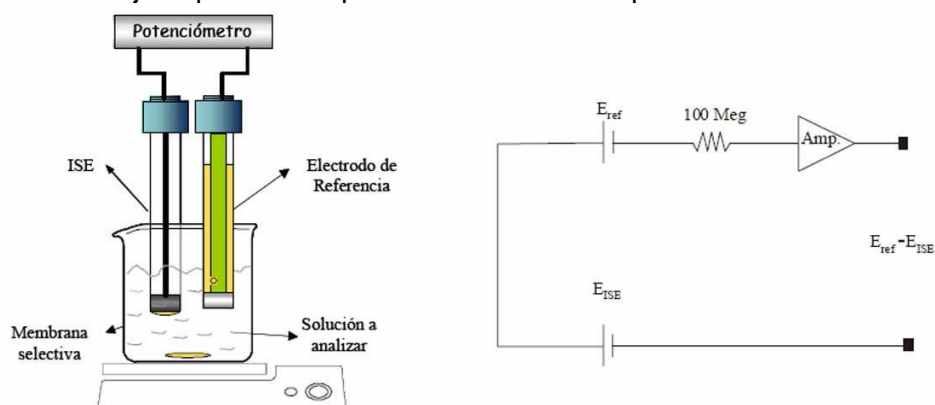
En los electrodos selectivos a iones (ISEs), la membrana es responsable tanto del reconocimiento selectivo como de la transformación del potencial generado en una señal eléctrica que se puede medir, es decir el receptor y el transductor están integrados en un único elemento.

En la Figura 15 se muestra el esquema eléctrico resultante para la medida del potencial eléctrico que se produce entre los dos electrodos.

¹⁷ **Quiminet.** Quiminet. " *Ventajas, usos y aplicaciones de las bombas peristálticas*". [En línea] [Citado el: 11 de Enero de 2010]. http://www.quiminet.com/ar0/ar_vcdRsDFaddsaaddsa-ventajas-usos-y-aplicaciones-de-las-bombas-peristalticas.htm

¹⁸ El cloruro mercurioso, también llamado calomelano o cloruro de mercurio es un compuesto inorgánico de fórmula Hg_2Cl_2 .

Figura 15. Montaje experimental para realizar medidas potenciométricas.



Fuente: Castillo, Omar y Villegas, Baryeric. Blogspot. "Sensor Potenciométrico". [En línea] [Citado el: 29 de Noviembre de 2009.] <http://sensorpotenciometrico.blogspot.com/>.

La diferencia de potencial medida es la suma de tres contribuciones: la interacción entre la membrana porosa y el ión de interés, la interacción que se origina entre el elemento de referencia y el medio donde está insertado, proporcionando un potencial constante si la composición del medio de referencia y la temperatura no varían y por último la denominada unión líquida, que se origina entre la solución de trabajo y la solución interna del electrodo de referencia. Los clásicos electrodos de referencia empleados en potenciómetro, calomelano y Ag/AgCl fijan su potencial utilizando una solución de cloruro de actividad constante. Frecuentemente es necesario el uso de una segunda disolución de electrolito que se interpone, mediante un puente salino, entre esta disolución interna de referencia y la muestra, con el fin de evitar la interacción entre ambas. Esta unión líquida representa una interfase donde un electrolito se difunde en el otro, siendo la causa del surgimiento de un potencial eléctrico, que contribuye al potencial de la celda, denominado potencial de unión líquida. Esta diferencia de potencial suele ser pequeña y normalmente es de magnitud desconocida.¹⁹

Por otra parte, debido a que el potencial de interfase interno y el potencial del electrodo de referencia interno son constantes, la medida del potencial que se genera en la celda electroquímica, representa únicamente el cambio de potencial que se produce a través de la superficie de la membrana del ISE, obteniéndose de esta forma información de la concentración de iones en la muestra.

¹⁹ Castillo, Omar y Villegas, Baryeric. Blogspot. "Sensor Potenciométrico". [En línea] [Citado el: 29 de Noviembre de 2009.] <http://sensorpotenciometrico.blogspot.com/>.

4.5.1 Tipos de Electrodo de Medición. Los sensores potenciométricos simplemente miden el potencial de equilibrio termodinámico y en el cual esencialmente no fluye corriente neta. El instrumental necesario para las medidas potenciométricas comprende un electrodo de referencia, un electrodo indicador y un dispositivo de medida de potencial; la construcción de los sensores potenciométricos depende del tipo de solución que se desea medir, y el tipo de electrodo que se utilice, ya sea el electrodo de referencia o el electrodo indicador.

4.5.1.1 Electrodo de Referencia. En muchas aplicaciones es deseable que el potencial de media celda de uno de los electrodos sea conocido, constante y completamente insensible a la composición de la solución en estudio. Un electrodo con estas características, se denomina electrodo de referencia. (¡Error! No se encuentra el origen de la referencia.)

Figura 16. Electrodo de Referencia.



Fuente: Castillo, Omar y Villegas, Baryeric. Blogspot. "Sensor Potenciométrico". [En línea] [Citado el: 29 de Noviembre de 2009.] <http://sensorpotenciometrico.blogspot.com/>.

Un electrodo de referencia debe ser fácil de montar, proporcionar potenciales reproducibles y tener un potencial sin cambios con el paso de pequeñas corrientes. Dos electrodos comúnmente utilizados que satisfacen estos requisitos son el Electrodo de Calomelano y el Electrodo de Plata-Cloruro de Plata.

4.5.1.2 Electrodo Indicador. Junto con el electrodo de referencia se utiliza un electrodo indicador cuya respuesta depende de la concentración del analito²⁰. Los electrodos indicadores para las medidas potenciométricas son de dos tipos fundamentales, denominados metálicos y de membrana. Estos últimos se denominan también electrodos específicos o selectivos para iones (Figura 17).

²⁰ En química analítica un analito es el componente (elemento, compuesto o ion) de interés analítico de una muestra. Son especies químicas cuya presencia o concentración se desea conocer.

Figura 17. Electrodo Indicadores.



Fuente: Castillo, Omar y Villegas, Baryeric. Blogspot. "Sensor Potenciométrico". [En línea] [Citado el: 29 de Noviembre de 2009.] <http://sensorpotenciometrico.blogspot.com/>.

4.5.2 Efectos de la temperatura sobre la medición de pH

Cuando se mide el pH de una sustancia en general ocurre que con un cambio de la temperatura la medida se vea afectada, esto se debe a que la pendiente del electrodo varía con la temperatura de forma conocida, según la ecuación de Nernst.

Ecuación 1. Ley de Nernst

$$E = E^0 - \frac{RT}{nF} \ln Q_c$$

Donde:

E = Potencial de la pila (V), en condiciones específicas.

E⁰ = Potencial de la pila en condiciones de estándar.

R = Constante de gas ideal = 8.314 J / mol·K .

T = Temperatura (Kelvin), que es generalmente de 25 °C (298 K) C (298 K).

n = número de moles de electrones transferidos en la ecuación de equilibrio.

F = Constante de Faraday, la carga de un mol de electrones = 95,484.56 C/mol

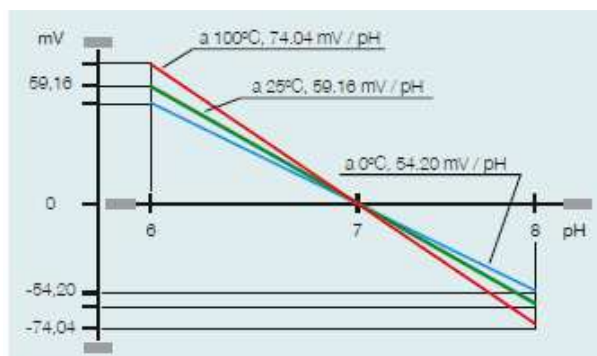
lnQ_c = El logaritmo natural del cociente de reacción en el momento en el tiempo.

Los transmisores de pH generalmente realizan la compensación automática a este efecto, esto lo de tres distintas formas:

- Conectar al transmisor un sensor de temperatura, además del electrodo.
- Utilizar un electrodo con sensor de temperatura incorporado cuya ventaja principal es que facilita la manipulación. Con un solo sensor y un solo cable se transmiten las dos señales, pH y temperatura, al instrumento.
- Introducir la temperatura manualmente. Hay aplicaciones donde no se requiere la medida de la temperatura, bien porque ésta es constante o bien porque se acepta una tolerancia amplia en el valor de pH, etc. En estos casos, la temperatura de la muestra se puede introducir manualmente mediante el teclado del instrumento.

En la Figura 18 se muestra el efecto de la temperatura sobre la pendiente del sensor.

Figura 18. Efecto de la temperatura en la pendiente del sensor.



En la Figura 19 se muestra el error cometido en la medida de pH sin compensación de la temperatura.

Figura 19. Error cometido en la medida de pH sin compensación de la temperatura.

°C	pH											
	2	3	4	5	6	7	8	9	10	11	12	
5	0.30	0.24	0.18	0.12	0.06	0	0.06	0.12	0.18	0.24	0.30	
15	0.15	0.12	0.09	0.06	0.03	0	0.03	0.06	0.09	0.12	0.15	
25	0	0	0	0	0	0	0	0	0	0	0	
35	0.15	0.12	0.09	0.06	0.03	0	0.03	0.06	0.09	0.12	0.15	
45	0.30	0.24	0.18	0.12	0.06	0	0.06	0.12	0.18	0.24	0.30	
55	0.45	0.36	0.27	0.18	0.09	0	0.09	0.18	0.27	0.36	0.45	
65	0.60	0.48	0.36	0.24	0.12	0	0.12	0.24	0.35	0.48	0.60	
75	0.75	0.60	0.45	0.30	0.15	0	0.15	0.30	0.45	0.60	0.75	
85	0.90	0.72	0.54	0.36	0.18	0	0.18	0.36	0.54	0.72	0.90	

Sin error Error < 0.1 pH Error > 0.1 y < 0.3 pH Error ≥ 0.3 pH

4.6 CONTROL INTELIGENTE

El incremento de las demandas tecnológicas en nuestros tiempos ha generado sistemas muy complejos que requieren controladores altamente sofisticados para asegurar un alto desempeño dentro de condiciones adversas, esto debido a que la complejidad del sistema y el hecho de que la estructura y los parámetros en muchos sistemas cambien de manera significativa e impredecible con el tiempo, lo cual dificulta la representación del sistema en un modelo matemático "simple". Es bajo estas condiciones en donde se utilizan las técnicas del control inteligente.

El control inteligente es la disciplina donde los métodos de control se desarrollan para emular algunas características importantes del ser humano, en las que se incluyen adaptación y aprendizaje, planeación bajo gran incertidumbre y el trabajo con gran cantidad de datos.

Dentro del control inteligente se encuentran principalmente tres tipos de controladores principales que son: control neuronal, control difuso y control neuro-difuso.

A continuación se explican algunos conceptos de gran importancia para comprender el funcionamiento de un controlador difuso.

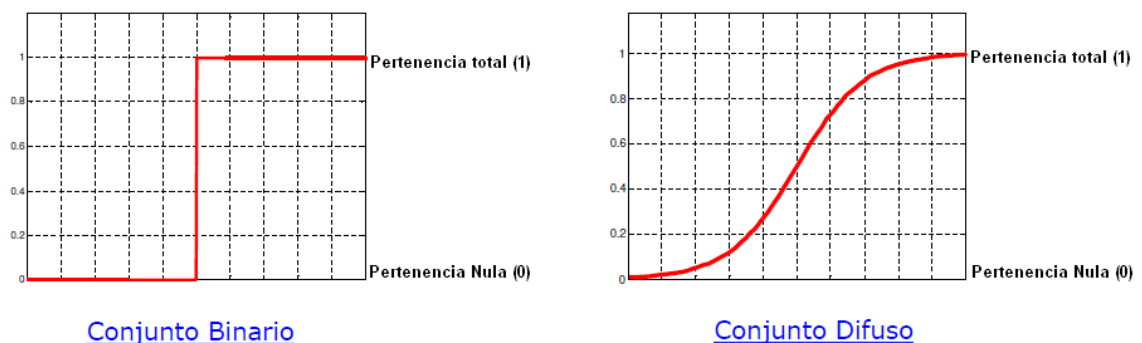
4.6.1 Lógica difusa. Una de las disciplinas matemáticas con mayor número de seguidores actualmente es la llamada lógica difusa o borrosa, que es la lógica que utiliza expresiones que no son ni totalmente ciertas ni completamente falsas, es decir, es la lógica aplicada a conceptos que pueden tomar un valor cualquiera de veracidad dentro de un conjunto de valores que oscilan entre dos extremos, la verdad absoluta y la falsedad total.

Conviene recalcar que lo que es difuso, borroso, impreciso o vago no es la lógica en sí, sino el objeto que estudia: expresa la falta de definición del concepto al que se aplica. La lógica difusa permite tratar información imprecisa, como *estatura media* o *temperatura baja*, en términos de conjuntos borrosos que se combinan en reglas para definir acciones: *si la temperatura es alta entonces enfriar mucho*. De esta manera, los sistemas de control basados en lógica difusa combinan variables de entrada, definidas en términos de conjuntos difusos, por medio de grupos de reglas que producen uno o varios valores de salida.

4.6.1.1 Conjuntos difusos. Formalmente, un conjunto difuso A se define como una función de pertenencia que enlaza o empareja los elementos de un dominio o universo de discurso X con elementos del intervalo $[0,1]$.

Podemos decir que la teoría clásica de conjuntos sólo contempla la pertenencia o no pertenencia de un elemento a un conjunto, a diferencia la teoría de conjuntos difusos que contempla la pertenencia parcial de un elemento a un conjunto, es decir, cada elemento presenta un grado de pertenencia a un conjunto difuso que puede tomar cualquier valor entre 0 y 1 (Figura 20). Este grado de pertenencia se define mediante la función característica asociada al conjunto difuso, lo cual quiere decir que para cada valor que pueda tomar un elemento o variable de entrada x , la función característica $\mu_A(x)$ proporciona el grado de pertenencia de este valor de x al conjunto difuso A .

Figura 20. Conjunto binario vs conjunto difuso.



4.6.1.2 Funciones de pertenencia. Las funciones de pertenencia $\mu(x)$ son las encargadas de definir a los conjuntos difusos. Cualquier función de pertenencia puede ser utilizada para definir un conjunto difuso, sin embargo, es preferible utilizar funciones “simples” debido a que simplifican muchos cálculos y no pierden exactitud.

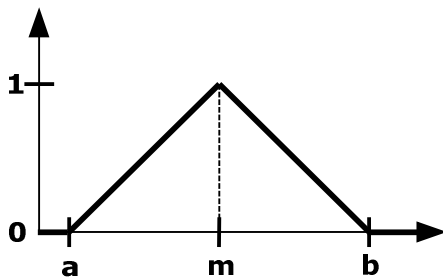
Las funciones de pertenencia más utilizadas son:

- Función triangular.
- Función Γ (gamma).
- Función sigmoidal.
- Función gaussiana.
- Función trapezoidal.

Función triangular:

Está definida por su límite inferior a , su límite superior b , y su valor modal m tal que $a < m < b$.

Figura 21. Función triangular.

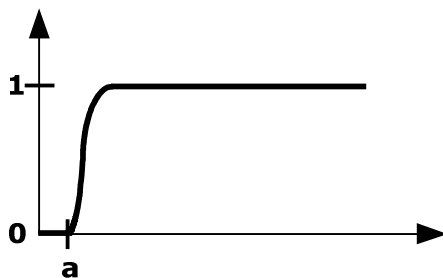


$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ (x - a)/(m - a) & \text{si } a < x \leq m \\ (b - x)/(b - m) & \text{si } m < x < b \\ 0 & \text{si } x \geq b \end{cases}$$

Función Γ (gamma):

Esta definida por su límite inferior a y el valor de la constante $k > 0$.

Figura 22. Función Γ .

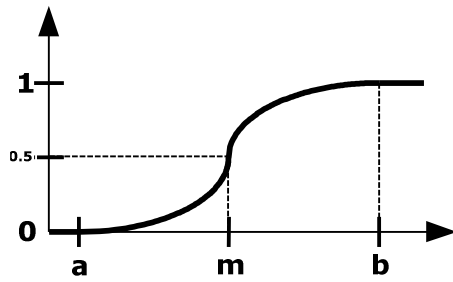


$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ 1 - e^{-k(x-a)^2} & \text{si } x > a \end{cases}$$

Función sigmoidal:

Definida por su límite inferior a , su límite superior b y el valor m o punto de inflexión tal que $a < m < b$.

Figura 23. Función sigmoidal

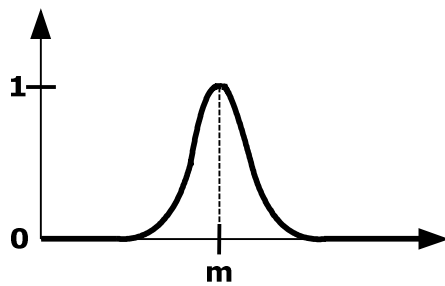


$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ 2[(x-a)/(b-a)]^2 & \text{si } a < x \leq m \\ 1 - 2[(x-a)/(b-a)]^2 & \text{si } m < x < b \\ 1 & \text{si } x \geq b \end{cases}$$

Función gaussiana:

Definida por su valor medio m y el valor $k > 0$

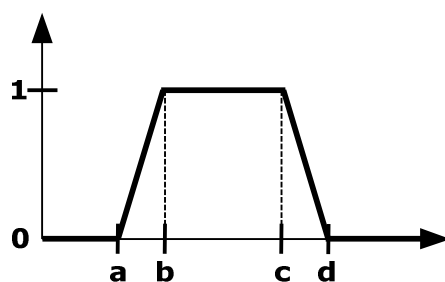
Figura 24. Función gaussiana.



$$\mu_A(x) = e^{-k(x-m)^2}$$

Función trapezoidal:

Figura 25. Función trapezoidal



$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ (x-a)/(b-a) & \text{si } a < x \leq b \\ 1 & \text{si } b < x < c \\ (d-x)/(d-c) & \text{si } c < x < d \\ 0 & \text{si } x \geq d \end{cases}$$

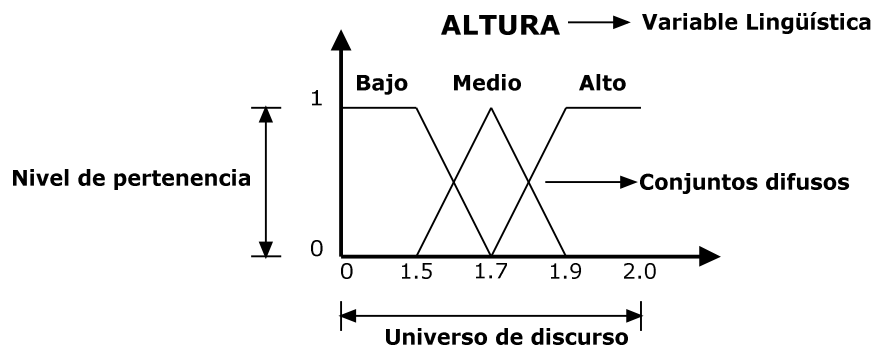
4.6.1.3 Variables lingüísticas. Una variable lingüística es aquella noción o concepto que calificamos de forma difusa, el adjetivo "lingüística" se aplica porque define sus características mediante el lenguaje hablado²¹. Por ejemplo: la altura, la edad, el error, la variación del error, etc.

4.6.1.4 Valor lingüístico. El valor lingüístico es una clasificación que efectuamos sobre una variable lingüística; en el caso de la altura, podríamos dividir el universo de discurso en los diferentes valores lingüísticos: 'bajo', 'mediano' y 'alto'. Cada valor lingüístico deberá tener un conjunto difuso asociado, de forma que hablaremos de los conjuntos difusos 'bajo', 'alto', etc... asociados a la variable lingüística 'altura'.

4.6.1.5 Universo de discurso. El universo de discurso X es el rango de valores que pueden tomar los elementos que poseen la propiedad expresada por la variable lingüística. En el caso de la variable lingüística 'altura', sería el conjunto de valores comprendido entre 1.5 y 2.3 m.

En la Figura 26 se representan todos los conceptos de lógica difusa mencionados anteriormente.

Figura 26. Conceptos de lógica difusa.



4.6.2 Controladores difusos

Para el diseño de un controlador difuso, como ya se mencionó anteriormente, no se requiere el modelo analítico completo del sistema dinámico. El resultado de este diseño es un controlador heurístico²² basado en conocimiento, utilizado para controlar un sistema complejo e indefinido.

²¹ La Salle. *Universidad Ramón Llull*. [En línea] [Citado el: 11 de 01 de 2010.] <http://www.salle.url.edu/portal/lasalle/enginyeria-arquitectura/Controller?mvchandler=portals&action=change-language&view=collage&lang=es>.

²² Se denomina heurística a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines

Figura 27. Esquema de un sistema retroalimentado de control clásico.

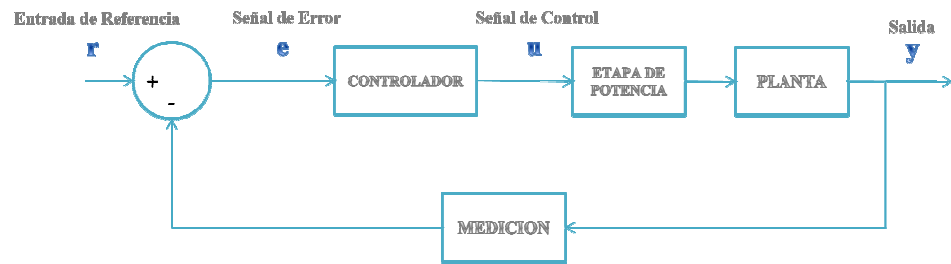
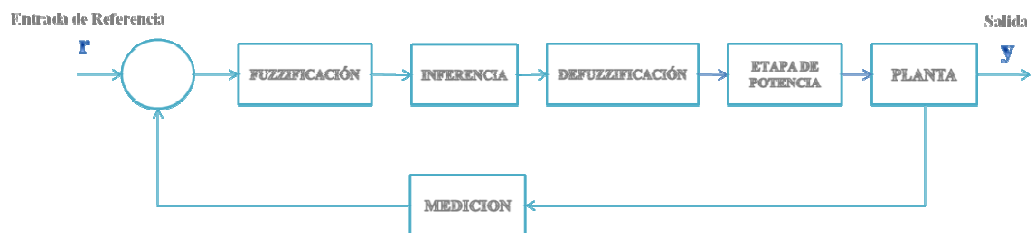


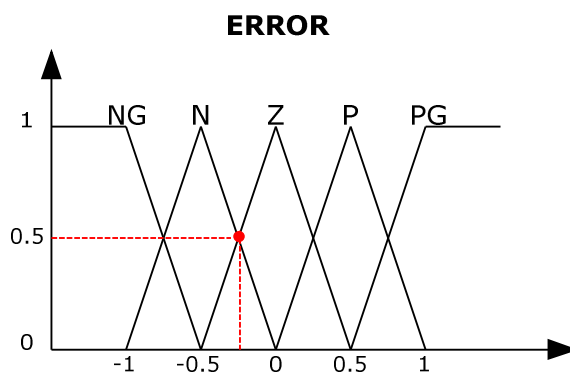
Figura 28. Esquema de controlador difuso



En el esquema de un controlador difuso mostrado en la Figura 28 aparecen tres bloques que no estaban presentes en el esquema de control clásico (Figura 27), uno etiquetado como Fuzzificación, otro como Defuzzificación y otro como Inferencia.

La Fuzzificación es el proceso de traducción de valores del mundo real al ambiente Fuzzy mediante el uso de funciones de pertenencia. Para comprender mejor todo lo planteado anteriormente analizaremos una situación aleatoria:

Figura 29. Ejemplo de una situación en un instante t.



La Figura 29 nos indica que en el instante t la variable error tiene un valor en el mundo real equivalente a -0.25 sin importar lo que eso signifique. Cuando el controlador fuzzifica esta situación el valor de la variable error se transforma de -0.25 a “0.5 perteneciente a N y 0.5 perteneciente a Z”.

Después de tener la situación descrita en términos lingüísticos el controlador realiza una inferencia teniendo en cuenta un conjunto de reglas previamente definidas por el diseñador del mismo; este razonamiento se hace sólo tomando las reglas que describen la situación actual, en otras palabras, las que tienen un nivel de pertenencia distinto de cero.

Las reglas deben ser escritas según la siguiente sintaxis:

IF precedente THEN consecuente.

Para explicar mejor lo mencionado anteriormente retomaremos la situación descrita en la Figura 29, sólo que ahora vamos a tener los siguientes conjuntos difusos haciendo referencia a la salida: Z, P, M, G y MG.

Suponiendo que el desarrollador definió las siguientes reglas:

1. IF Error is NG THEN Salida is MG
2. IF Error is N THEN Salida is G
3. IF Error is Z THEN Salida is Z
4. IF Error is P THEN Salida is Z

Dado que el error en el instante t sólo tiene pertenencia en los conjuntos N y Z, solo se van a involucrar las reglas que tienen estas situaciones en el precedente, en este caso la dos y la tres.

Sin embargo, hasta ahora no tenemos un valor en el mundo real que nos representa la respuesta que el control está dando para la situación actual, esto es porque este proceso está relacionado con el módulo Defuzzificación.

Para realizar la defuzzificación de un sistema de control difuso existen dos opciones las cuales darán nombre a nuestro sistema, pues dependiendo de que método utilicemos el sistema será de tipo "Mamdani" o "Takagi Sujeno". En la actualidad ambos son igualmente populares, sin embargo cuando se quiere programar un controlador difuso en un dispositivo de gama media como un microcontrolador PIC16xxx o incluso en un PIC18xxx, resulta mucho más cómodo trabajar con sistemas tipo Takagi Sujeno ya que ciertamente facilita muchos cálculos.

Un sistema tipo Takagi Sujeno realiza la Defuzzificación de la salida de acuerdo con la siguiente expresión:

Ecuación 2. Defuzzificación de un sistema difuso tipo Takagi Sujeno.

$$y = \frac{w_1 f_1 + w_2 f_2 + \dots + w_n f_n}{w_1 + w_2 + \dots + w_n}$$

Donde:

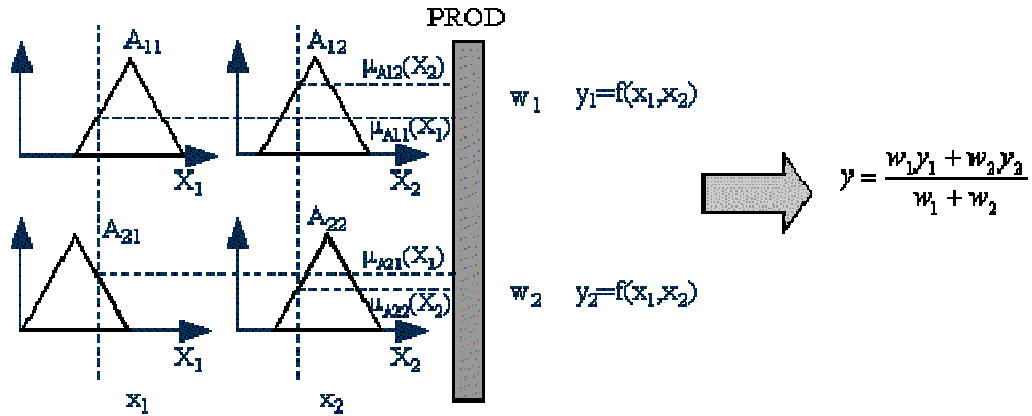
w_n es el nivel de pertenencia de la regla n .

f_n es la función asociada a la salida de la regla.

y es la salida del controlador en el mundo real.

En la Figura 30 se muestra detallado el funcionamiento de un sistema tipo Sujeno²³:

Figura 30. Funcionamiento de un sistema difuso tipo Takagi Sujeno.



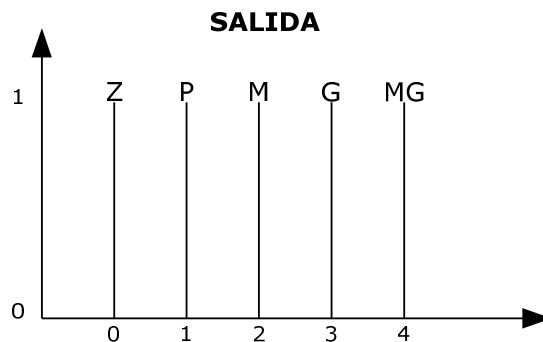
Fuente: López, Jesús Alfonso. Tripod. [En línea] [Citado el: 16 de 01 de 2010.] http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html.

En caso de que una regla tenga más de un conjunto asociado en el consecuente el nivel de pertenencia de la regla estará dado por la multiplicación de los niveles de pertenencia de dichos conjuntos.

En un sistema Sujeno la salida está representada por funciones o incluso por constantes, para éste último caso el conjunto de salida tiene el nombre de "Singleton".

Para aclarar más los conceptos anteriores, se va a terminar el ejemplo de la Figura 29, para esto se definirá el universo de discurso de la salida con sus respectivas funciones:

Figura 31. Representación de la salida de un controlador difuso con funciones Singleton.



²³ López, Jesús Alfonso. Tripod. [En línea] [Citado el: 16 de 01 de 2010.] http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html.

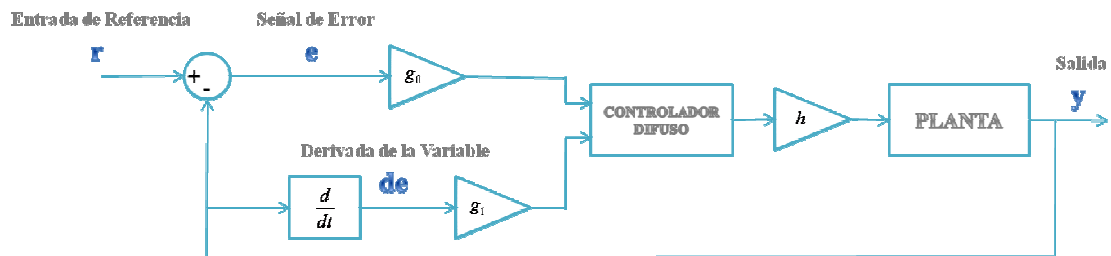
Según lo que habíamos planteado el controlador solo toma las reglas:

IF Error is N THEN Salida is G
IF Error is Z THEN Salida is Z

Aplicando la Ecuación 2 obtenemos:

4.6.2.1 Esquemas de control difuso. Un esquema de control regulatorio difuso generalmente tiene la forma mostrada en la Figura 32.

Figura 32. Esquema para la implementación de un controlador difuso.



Fuente: Scribd. *Control Difuso*. [En línea] [Citado el: 01 de 02 de 2010.] <http://www.scribd.com/doc/3200501/Control-Difuso>.

Los controladores difusos se pueden sintonizar cambiando el valor de las ganancias h , g_0 y g_1 . A continuación se explica el efecto de cada ganancia sobre el controlador²⁴:

Efectos según el valor de g_0 :

Si $g_0=1$, no tiene ningún efecto sobre el sistema.

Si $g_0<1$, los conjuntos difusos del universo de discurso del error son uniformemente expandidos en el eje horizontal por un factor de $1/g_0$.

Si $g_0>1$, los conjuntos difusos del universo de discurso del error son uniformemente contraídos en el eje horizontal por un factor de $1/g_0$.

Efectos según el valor de g_1 :

Si $g_1=1$, no tiene ningún efecto sobre el sistema.

Si $g_1<1$, los conjuntos difusos del universo de discurso de la derivada de la variable son uniformemente expandidos en el eje horizontal por un factor de $1/g_1$.

²⁴ **Pacino, Kevin M.** *Fuzzy Control*. Menlo Park, California : Addison-Wesley, 1988. págs. 86-87. ISBN 0-201-18074-X.

Si $g_1 > 1$, los conjuntos difusos del universo de discurso de la derivada de la variable son uniformemente contraídos en el eje horizontal por un factor de $1/g_1$.

Efectos según el valor de h :

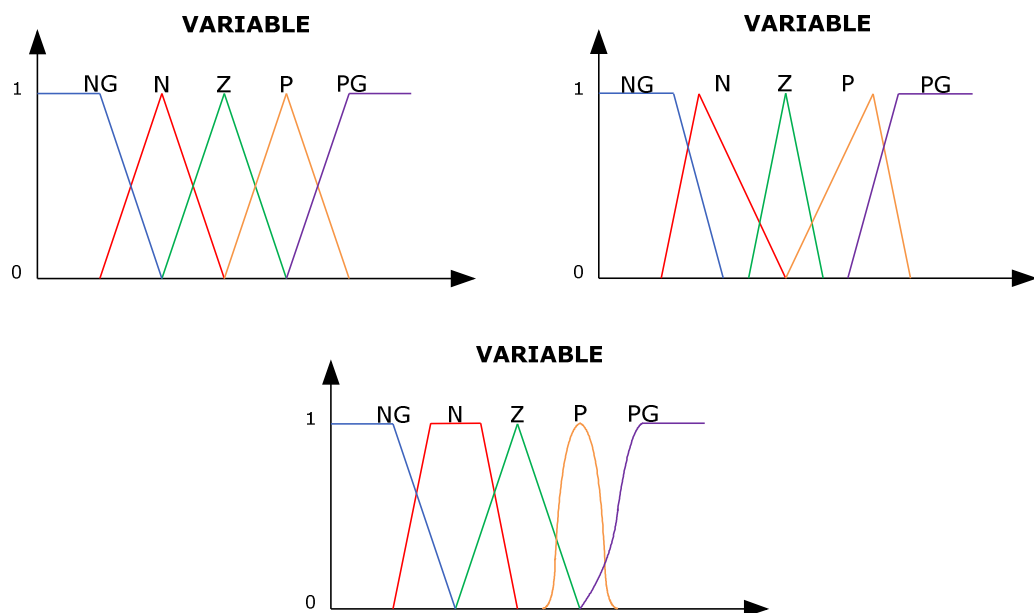
Si $h=1$, no tiene ningún efecto sobre el sistema.

Si $h < 1$, los conjuntos difusos del universo de discurso de la salida son uniformemente expandidos en el eje horizontal por un factor de h .

Si $h > 1$, los conjuntos difusos del universo de discurso de la salida son uniformemente contraídos en el eje horizontal por un factor de h .

También se puede mejorar el rendimiento en la operación del controlador cambiando los conjuntos que conforman el universo de discurso del error y la derivada de la variable como se muestra en la Figura 33.

Figura 33. Sintonización por medio de conjuntos difusos.



4.6.2.2 Algoritmos para realizar controladores difusos. El siguiente algoritmo permite crear un controlador difuso según el esquema mostrado en la Figura 32.

1. Obtener los valores del error y la derivada del error.
2. Calcular el nivel de pertenencia de cada variable a cada conjunto de su respectivo universo de discurso.
3. Calcular el nivel de pertenencia de cada regla.
4. Realizar la defuzzificación según el método planteado en la Ecuación 2.

En la descripción del diseño detallado del dispositivo se encuentra un código implementando el algoritmo anteriormente mencionado debidamente explicado.

4.7 REGULACIÓN POR ÁNGULO DE FASE

La regulación por ángulo de fase se basa en la posibilidad, con la tecnología electrónica existente, de poder realizar la conexión de la tensión de red en cualquier punto de la semionda de forma sincronizada. De esta forma se puede «recortar» la tensión sinusoidal en sectores mas pequeños, lo que permite, al fin, regular la potencia aplicada a la carga.

Considerando que cada semionda sinusoidal de red corresponde a una conducción de 180° (360° para una onda completa), retardando la conexión de 180° a 0° conseguimos regular la potencia aplicada a la carga del 0 al 100%, a este valor de retardo se le llama «ángulo de retardo» y se indica normalmente con la letra α (alfa). En algunos casos se habla también del «ángulo de conducción», que corresponde al valor $180-\alpha$.

Visto desde otra forma se puede ver como un control PWM aplicado a la corriente altera, salvo que la relación entre el voltaje RMS y el ángulo de conducción no será lineal.

4.7.1 Tensión eficaz V_{RMS} . Por definición, el valor RMS de una señal AC es el valor el que produce el mismo efecto calorífico que su equivalente en DC. En otras palabras y a modo de ejemplo ilustrativo, si a un resistor le conectamos una tensión continua de 220Volts, dicho resistor disipará la misma cantidad de energía que si le conectamos una tensión alterna de 220Volts eficaces o RMS.

4.7.2 Relación entre la tensión eficaz V_{RMS} y el ángulo de retardo. Matemáticamente, el valor eficaz de una señal variable con el tiempo, se define como la raíz cuadrada del valor medio de la señal elevado al cuadrado alcanzado durante un período. Esto es:

Ecuación 3. Definición matemática del valor eficaz de la tensión de una señal.

$$V_{RMS} = \sqrt{\frac{V_p}{T} \int_0^T f(x)^2 dx}$$

4.7.3 Relación entre la tensión eficaz V_{RMS} y el ángulo de retardo. Dado que en la regulación de ángulo de fase regulamos el tramo de la onda que recibe la carga resistiva, el valor RMS que recibe la carga cuando el tramo de la señal se encuentra entre α y 180° es:

Ecuación 4. Definición matemática del valor eficaz de la tensión de una señal según el ángulo de disparo.

$$V_{RMS} = \sqrt{\frac{1}{T} \int_{\alpha}^T V_p^2 f(x)^2 dx}$$

Si que función que describe el comportamiento de la señal es $f(x)=\text{sen}(x)$ y nuestro periodo de interés es un semiperiodo de la onda en otras palabras 180° , entonces al resolver la integral obtenemos que:

$$V_{RMS} = \sqrt{\frac{V_p^2}{180} \left[\frac{180 - \alpha}{2} - \frac{\text{sen}(180 - \alpha) \cos(180 - \alpha)}{2} \right]}$$

Al simplificar la expresión tenemos que:

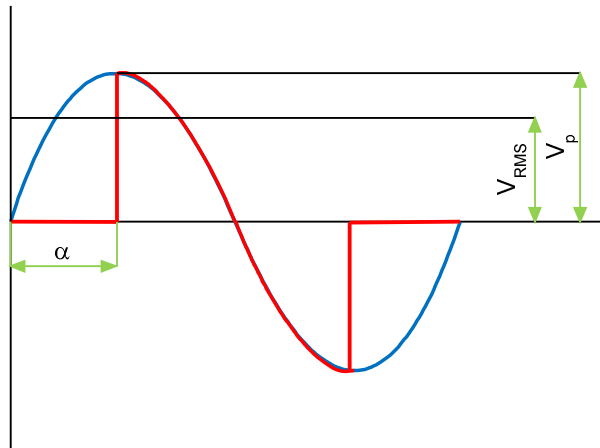
Ecuación 5. Relación entre la tensión eficaz y el ángulo de disparo para el semiperiodo de la señal senoidal.

$$V_{RMS} = \frac{V_p}{\sqrt{360}} \sqrt{180 - \alpha + \frac{1}{2} \sin(2\alpha)}$$

Donde el valor de α está dado en grados.

Debido a que en la expresión obtenida en la Ecuación 5 contiene el término α dentro y fuera de una señal tipo senoidal, es imposible hallar el ángulo de disparo en términos de la tensión eficaz, por ende comúnmente se realiza una tabla donde se muestre el ángulo requerido para una tensión eficaz deseada.

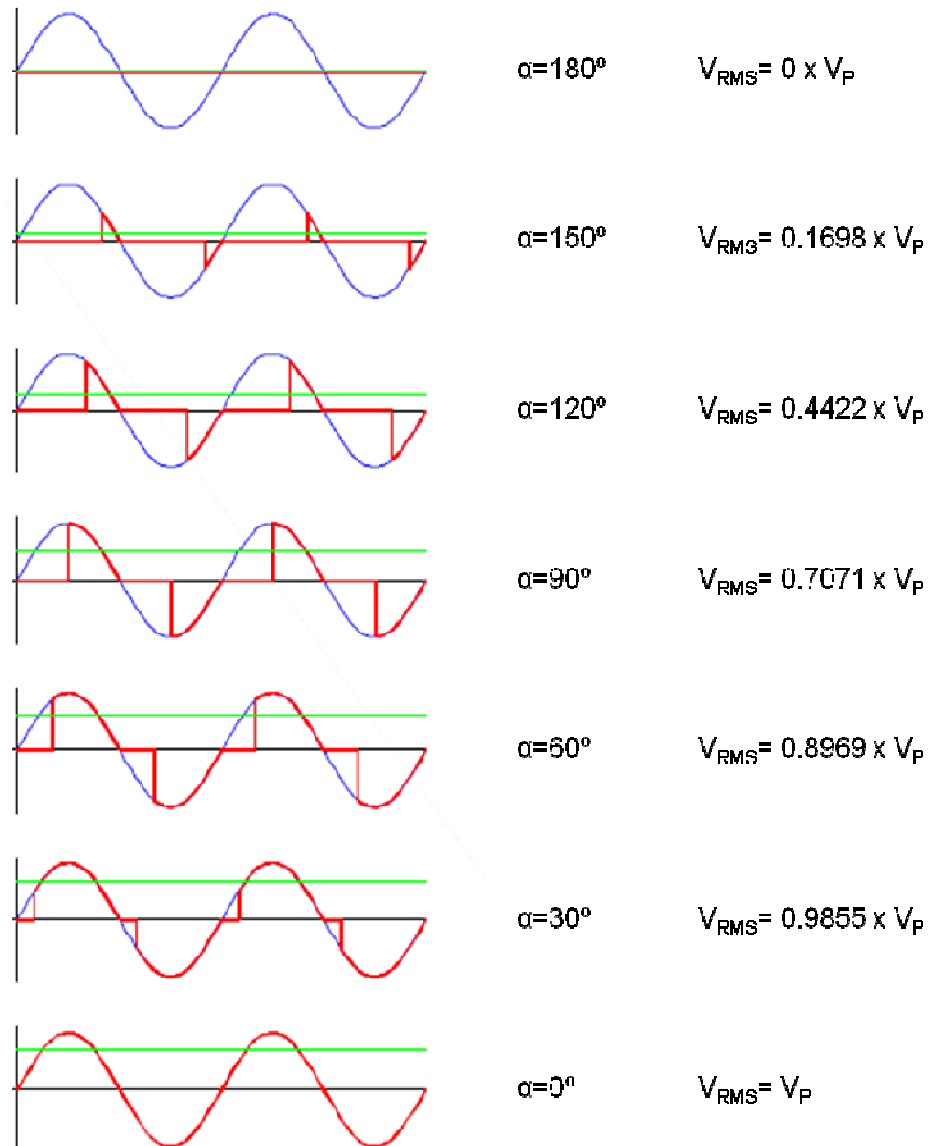
Figura 34. Efecto regulatorio del control por ángulo de fase.



El efecto regulatorio del control por ángulo de fase se puede observar en la Figura 34, en la que la onda azul representa la señal de entrada y la señal roja representa la señal con el efecto de regulación aplicado.

A continuación siguiente podemos ver la forma de onda y la relación de los valores ERMS respecto a diversos ángulos de retardo.

Figura 35. Relación entre la tensión pico y la tensión eficaz para ángulos de conducción comúnmente utilizados.



Fuente: GUASCH. *Componentes y electrónica de potencia*. [En línea] [Citado el: 02 de 01 de 2010.] http://www.e-guasch.com/onlinedocs/catalogue/appnotes/AN080917_CONTROL%20VS%20ANGULO%20DE%20FASE%20EN%20REGULADORES%20AC.pdf.

5 CAPITULO 1. Desarrollo de las etapas del diseño concurrente para el sistema de control de temperatura y pH del biorreactor de laboratorio

Introducción

El diseño concurrente o diseño simultáneo es una integración sistemática en el proceso de diseño, en la cual, los desarrolladores tienen en cuenta desde el inicio todos diferentes enfoques implicados en la elaboración del producto, entre ellos se encuentran: despliegue de la función de calidad (DFQ), diseño para funcionamiento (DFF), diseño para la manufactura (DFM), diseño para ensamble (DFA), y diseño para el mantenimiento (DFMT).

5.1 Desarrollo conceptual

En esta etapa se desarrolla el análisis funcional del producto identificando sus funciones y sub-funciones principales, se evalúan los conceptos seleccionados y adicionalmente se realizan algunos cálculos de ingeniería necesarios para el desarrollo de éstos conceptos.

5.1.1 Identificación de las necesidades. De acuerdo al análisis del proceso y a las entrevistas realizadas a los clientes, las cuales tenían como objetivo permitir conocer las necesidades que ellos tenían, se realizó la Tabla 5.

Tabla 5. Identificación de las necesidades.

PLANTEAMIENTOS DE LOS CLIENTES	IDENTIFICACIÓN DE LAS NECESIDADES
1. "El dispositivo debe ser capaz de controlar automáticamente la temperatura del producto con un margen de error máximo de ± 2 °C".	El dispositivo controlará automáticamente la temperatura del producto con margen de error de ± 2 °C.
2. "El dispositivo debe ser capaz de controlar automáticamente el nivel de pH del producto con margen de error máximo de ± 0.2 ".	El dispositivo controlará automáticamente el nivel de pH del producto con un margen de error máximo de ± 0.2 .
3. "Sin importar que ácido, base y tamaño de muestra que utilice el controlador de pH deberá seguir la referencia".	El controlador de pH deberá seguir la referencia sin importar que tipo de solución ácida/básica utilice y sin importar cantidad de la muestra".

Tabla 5. (Continuación)

PLANTEAMIENTOS DE LOS CLIENTES	IDENTIFICACIÓN DE LAS NECESIDADES
4. “No debe ser necesario apagar el dispositivo para cambiar la referencia de las variables”.	El dispositivo tendrá la opción de realizar el cambio de la referencia de ambas variables durante el proceso sin que requiera apagarse.
5. “El dispositivo debe ser lo más económico posible”.	El dispositivo será diseñado con el fin de ser lo más económico posible.
6. “No debe ser necesario tener un computador para operar el dispositivo”.	El dispositivo funcionará sin necesidad de estar conectado a un computador
7. “El dispositivo debe ser fácil de manipular para los operarios”.	El modo de operación será diseñado de forma que sea de fácil de operar.
8. “El dispositivo debe tener alarmas en caso de alguna falla”.	El dispositivo tendrá alarmas en caso de alguna falla.
9. “Las dimensiones del dispositivo final deben ser reducidas pues este debe ocupar el menor espacio posible”.	El dispositivo final será lo mas pequeño posible.
10. “El dispositivo debe darme la opción de controlar el nivel pH solo al inicio del proceso, o durante todo el proceso”.	El dispositivo tendrá la opción de controlar el nivel pH solo al inicio del proceso, o durante todo el proceso.
11. “Para el dispositivo de sebe usar la resistencia en abrazadera que tiene CLAYUCA”.	El dispositivo usará la resistencia tipo abrazadera que se encuentra en CLAYUCA.

Tabla 6. Métricas de las necesidades.

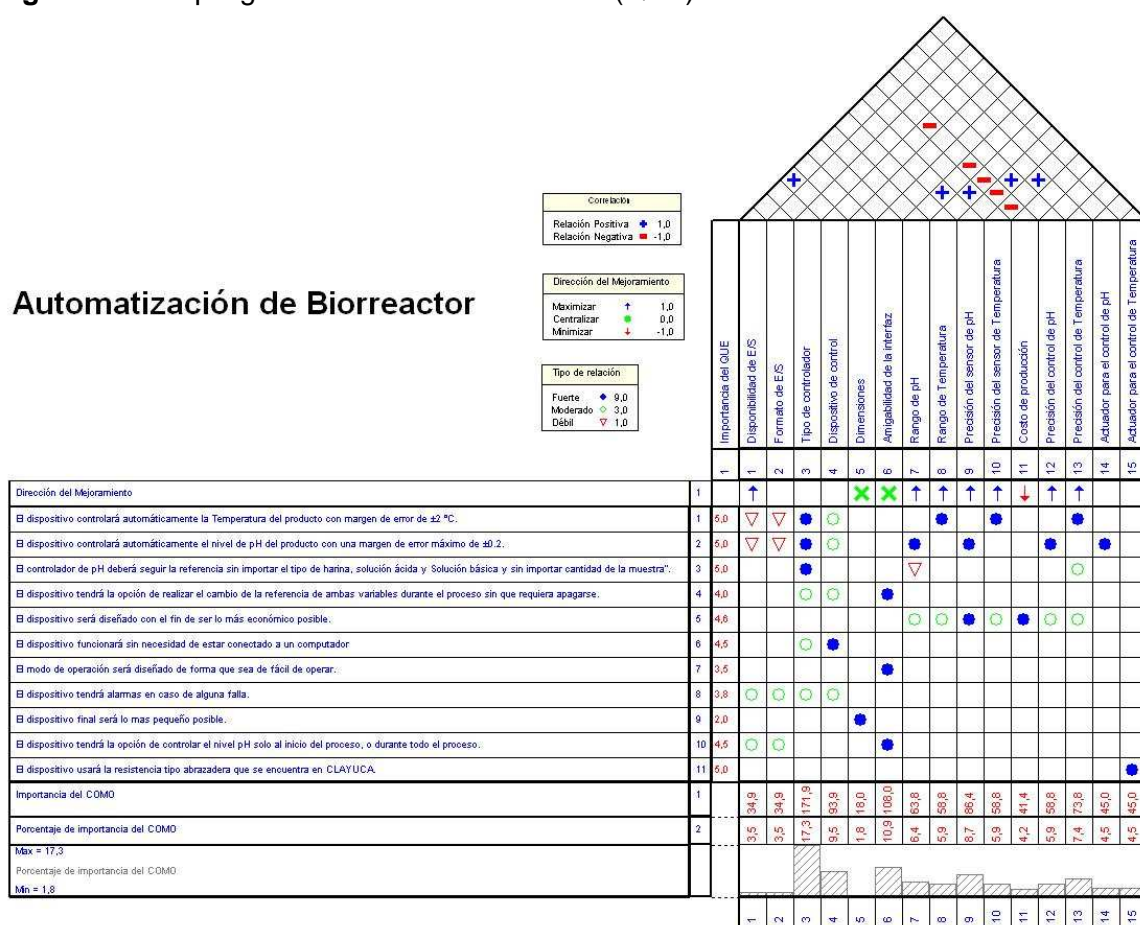
#	NECESIDAD	ESPECIFICACIONES TÉCNICAS	IMP	UNIDAD
1	1,2,8	Disponibilidad de E/S	4	#
2	1,2,8	Formato de E/S	4	Lista
3	1,2,3,4,6,8	Tipo de controlador	5	Lista
4	1,2,4,6,8	Dispositivo de control	5	Lista
5	9	Dimensiones	3	cm
6	4,7,10	Amigabilidad de la interfaz	3	Cualitativo
7	3,2	Rango de pH	4	#
8	1	Rango de Temperatura	4	°C
9	2	Precisión del sensor de pH	5	±#
10	1	Precisión del sensor de Temperatura	5	±°C
11	5	Costo de construcción	5	Col. \$
12	2	Precisión del control de pH	5	±#

Tabla 6. (Continuación).

#	NECESIDAD	ESPECIFICACIONES TÉCNICAS	IMP	UNIDAD
13	1	Precisión del control de Temperatura	5	±°C
14	2	Actuador para el control de pH	5	Lista
15	1,11	Actuador para el control de Temperatura	5	Lista

5.1.2 Establecer especificaciones preliminares

Figura 36. Despliegue de la función de calidad (QFD).



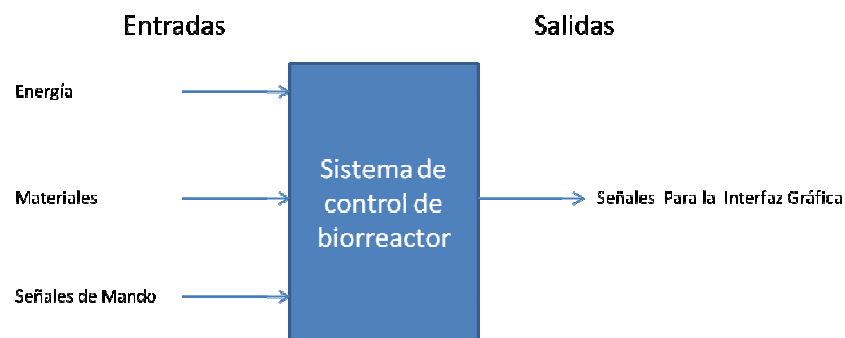
Especificaciones técnicas preliminares:

- Disponibilidad de entradas analógicas: 3 – 5.
- Disponibilidad de entradas digitales: 2 – 3.
- Disponibilidad de salidas analógicas: 0 - 1.
- Disponibilidad de salidas digitales: 18 – 26.
- Dispositivo de control: PIC16xx, PIC18xx, PLC (económico).
- Tipo de controlador: PID, control con estimador de estados, control PID difuso.
- Amigabilidad de la Interfaz: Cómoda – Muy cómoda.
- Rango de pH: 0 – 14.
- Rango de Temperatura: 0 °C – 100 °C.
- Precisión del sensor de pH: ±0.1 - ±0.2

- Precisión del sensor de Temperatura: $\pm 1\text{ }^{\circ}\text{C}$ - $\pm 2\text{ }^{\circ}\text{C}$.
- Costo de producción: \$1.000.000 – 2.500.000.
- Precisión del control de pH: ± 0.1 - ± 0.2
- Precisión del control de Temperatura: $\pm 1\text{ }^{\circ}\text{C}$ - $\pm 2\text{ }^{\circ}\text{C}$.
- Tipo de actuador de pH: Bombas dosificadoras, Bombas peristálticas.
- Tipo de actuador temperatura: Resistencia tipo abrazadera.

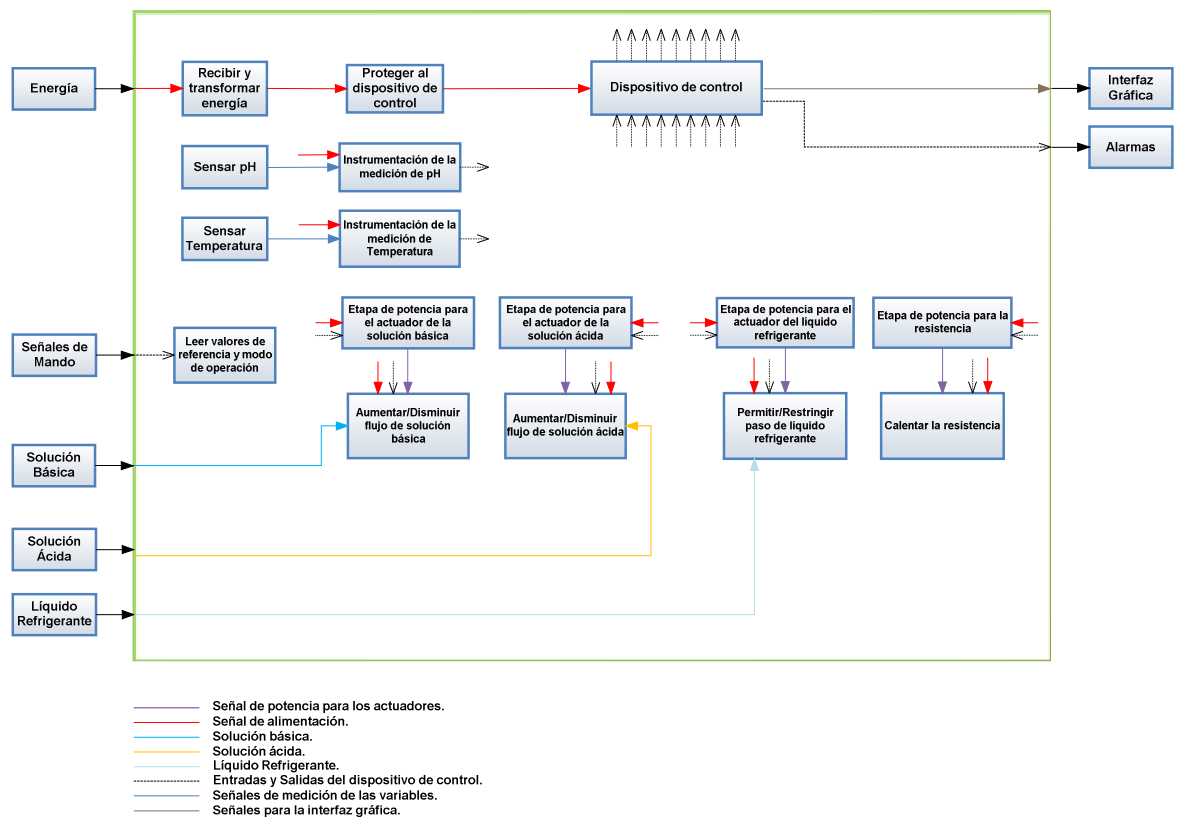
5.1.3 Generar conceptos. Viendo el proceso como un dispositivo de entradas y salidas sin importar los procesos internos del mismo se tiene la representación mostrada en la Figura 37.

Figura 37. Representación de las entradas y salidas del sistema de control del biorreactor.



Teniendo en cuenta este esquema general hacemos una descomposición funcional del sistema de control, y obtenemos lo mostrado en la Figura 38.

Figura 38. Descomposición funcional del sistema del control.



5.1.4 Seleccionar conceptos. Cada subfunción del diagrama de descomposición funcional representa la selección de un concepto, por lo tanto a cada subfunción se le hace un estudio de los dispositivos que cumplan con las especificaciones y al mismo tiempo hagan que el proceso sea eficiente.

Recibir y transformar energía:

Esta subfunción es la encargada de suministrar energía a todos los dispositivos que requieran de ella dentro de la descomposición funcional.

- Celdas solares.
- Energía eólica.
- Energía eléctrica tomada de la pared.

Proteger al dispositivo de control:

En esta subfunción se construirán los circuitos de protección que impedirán que el dispositivo de control sufra algún daño por sobre-voltaje o sobre-corriente.

Dispositivo de control:

Este dispositivo es el encargado de controlar tomar decisiones y controlar el proceso.

- PIC 16f877.
- PIC 18f4520.

Sensar pH:

Esta subfunción es la encargada de sensar el pH de la muestra que se está sometiendo al proceso de licuefacción y sacarificación.

- Sensor EPH-M11-Pt100-HT (Desin Instruments).
- Sensor Ap301/21020004 (ABB).

Sensar temperatura:

Esta subfunción es la encargada de sensar el pH de la muestra que se está sometiendo al proceso de licuefacción y sacarificación.

- Sensor EPH-M11-Pt100-HT (Desin Instruments).
- Sensor Ap301/21020134 (ABB).

Instrumentación de la señal de pH:

Esta subfunción es la encargada tomar la señal entregada por el sensor de pH, y entregarla al dispositivo de control en alguno de los formatos estándar (4 – 20 mA, 0 – 5 V, etc.)

- Transmisor DMM-4000-pH (Desin Instruments).
- Transmisor PHTX-271-1P (Omega).

Instrumentación de la señal de temperatura:

Esta subfunción es la encargada tomar la señal entregada por el sensor de temperatura, y entregarla al dispositivo de control en alguno de los formatos estándar (4 – 20 mA, 0 – 5 V, etc.)

- Transmisor DMM-4000-pH (Desin Instruments).
- Transmisor PHTX-271-1P (Omega).

Permitir restringir flujo de solución ácida/básica:

Estos dispositivos son aquellos encargados de introducir las soluciones ácidas o básicas para modificar el pH de la muestra.

- Bombas peristálticas SR-25-S300 20252201 (Thomas Industries).
- Bombas peristálticas M045 D.C. Tubo Ø1.6 mm (Verderflex).

Etapas de potencia de los actuadores pH:

Este módulo será el encargado transformar la señal entregada por el microcontrolador en una adecuada para ser aceptada por el actuador.

- Circuito compuesto de opto acopladores y transistores de potencia.
- Circuito integrado L298

Para la evaluación de los conceptos preliminares se ha establecido una lista de criterios mostrados a continuación:

Cantidad de E/S: La escala hace referencia a la cantidad de entradas y salidas que posee el dispositivo, donde 5 significa un alto número de entradas y salidas.

Compatibilidad: La escala hace referencia a la afinidad existente del dispositivo con otros ya implementados o en etapa de selección, en la que 5 significa un alto nivel de compatibilidad.

Consumo energético: La escala hace referencia a la cantidad de energía requerida por el dispositivo para funcionar por hora, siendo 5 poca energía necesaria.

Costo: La escala hace referencia al costo del dispositivo, donde 5 significa un costo bajo.

Disponibilidad de adquisición: La escala hace referencia a la facilidad de adquisición del dispositivo en Colombia, siendo 5 un dispositivo que cuenta con proveedores en el país.

Eficiencia: La escala hace referencia a la habilidad del dispositivo de responder y actuar adecuadamente, en la que 5 significa un alto nivel de eficiencia.

Facilidad de implementación: La escala hace referencia a la facilidad en el montaje estructural del dispositivo, teniendo en cuenta que ya existe una infraestructura, por ello 5 es el dispositivo que menos cambios requiere en esta infraestructura para el montaje.

Garantía: La escala hace referencia al tiempo y cobertura de la garantía, donde 5 indica una garantía ideal.

Mantenimiento: La escala hace referencia a la cantidad de horas trabajadas antes de que se requiera la atención del dispositivo y la complejidad al realizarlo, siendo 5 poca intervención y fácil ejecución del mantenimiento.

Seguridad: La escala referencia a los riesgos que genera la utilización e implementación del mismo, en la que 5 significa un alto nivel de seguridad.

Cada concepto será evaluado solamente con los criterios que sean relevantes para su operación.

5.1.4.1 Evaluación del concepto “Recibir y transformar energía”.

Los criterios para la evaluación de este concepto así como su importancia relativa en la evaluación se muestran en la Tabla 7.

Tabla 7. Criterios y nivel de importancia para la selección del concepto “Recibir y transformar energía”.

Criterio	Nivel de importancia
Seguro	5
Eficiencia	4
Mantenimiento	4
Costo	4
Disponibilidad de adquisición	5
Facilidad de implementación	4

En la Tabla 8 se encuentran los conceptos que van a ser evaluados según los criterios relevantes para cumplir la subfunción que tienen asignada.

Tabla 8. Selección del concepto “Recibir y transformar energía”.

Criterio para la selección del dispositivo	Importancia (1 - 5)	Celdas solares.		Energía eólica.		Energía eléctrica tomada de la pared.	
		Cal	Tot	Cal	Tot	Cal	Tot
Seguro	5	5	25	5	25	5	25
Compatibilidad	4	5	20	5	20	5	20
Eficiencia	4	4	16	3	12	5	20
Mantenimiento	4	2	8	3	12	5	20
Costo	4	2	8	3	12	5	20
Disponibilidad de adquisición	5	3	15	3	15	5	25
Facilidad de implementación	4	3	12	3	12	5	20
Total		24	104	25	108	35	150
Dispositivo seleccionado						X	

5.1.4.2 Evaluación del concepto “Dispositivo de control”

Figura 39. PIC18f4520.



Fuente: HVW Technologies. *PIC 16F877A/P*. [En línea] [Citado el: 10 de 01 de 2010.] http://www.hvwtech.com/products_view.asp?ProductID=883.

Información Técnica

Microcontrolador PIC18F4520

Fabricante:	Microchip
# de pines:	40
Memoria para código:	32 KB
Memoria para datos:	1536 B
Modulo CCP:	1 (CCP mejorado)
Comparador análogo:	1 (Dos entradas)
Timers:	4
Puertos de E/S	A-B-C-D-E
Comunicación Serial:	MSSP - EUSART
Precio:	\$ 18.000 COP

Figura 40. PIC16f877.



Fuente: HVW Technologies. *PIC 18F4520-I/P*. [En línea] [Citado el: 10 de 01 de 2010.] http://www.hvwtech.com/products_view.asp?ProductID=883.

Información Técnica

Microcontrolador PIC18F877A

Fabricante:	Microchip
# de pines:	40
Memoria para código:	8 KB
Memoria para datos	368 B
Modulo CCP:	2
Comparador análogo:	No
Timers:	3
Puertos de E/S	A-B-C-D-E
Comunicación Serial:	MSSP - EUSART
Precio:	\$ 10.000 COP

Los criterios para la evaluación de este concepto así como su importancia relativa en la evaluación se muestran en la Tabla 9.

Tabla 9. Criterios y nivel de importancia para la selección del concepto “Dispositivo de control”.

Criterio	Nivel de importancia
Cantidad de E/S	5
Compatibilidad	4
Eficiencia	4
Costo	4
Disponibilidad de adquisición	5
Consumo energético	5
Facilidad de implementación	4

En la Tabla 10 se encuentran los conceptos que van a ser evaluados según los criterios relevantes para cumplir la subfunción que tienen asignada.

Tabla 10. Selección del concepto “Dispositivo de Control”.

Criterio para la selección del dispositivo	Importancia (1 - 5)	PIC18f4520		PIC16f877	
		Cal	Tot	Cal	Tot
Seguro	5	4	20	4	20
Compatibilidad	4	5	20	5	20
Eficiencia	4	5	20	3	12
Consumo energético	5	5	25	5	25
Costo	4	5	20	5	20
Disponibilidad de adquisición	5	5	25	5	25
Facilidad de implementación	4	4	16	3	12
Total		33	146	30	134
Dispositivo seleccionado		X			

5.1.4.3 Evaluación del concepto “Sensor pH”

Figura 41. Electrodo EPH-M11-Pt100-HT.



Fuente: Design. *Electrodos industriales pH - ORP*. [En línea] [Citado el: 03 de 01 de 2010.] <http://www.desin.com/principal.asp>.

Electrodo EPH-M11-Pt100-HT

Fabricante:	Desin Instruments
Cuerpo:	PPS
Precisión de la medición:	±0,1% en todo el rango
Temperatura del medio:	0 - 110 °C
Compensación de temperatura	Pt100
Costo:	\$ 362.000 COP

Información Técnica AP301/2120134

Figura 42. Electrodo AP301-2120134.



Electrodo AP301/2120134

Fabricante:	ABB
Cuerpo:	PPS
Rango de pH:	0 - 14 pH
Precisión de la medición:	±0,1% en todo el rango
Temperatura del medio:	0 - 110 °C
Compensación de temperatura	Pt100
Costo:	\$ 521.000 COP

Los criterios para la evaluación de este concepto así como su importancia relativa en la evaluación se muestran en la Tabla 11.

Tabla 11. Criterios y nivel de importancia para la selección del concepto “Sensor pH”.

Criterio	Nivel de importancia
Compatibilidad	5
Eficiencia	5
Consumo energético	5
Costo	5
Disponibilidad de adquisición	5
Facilidad de implementación	4
Garantía	5

En la Tabla 12 se encuentran los conceptos que van a ser evaluados según los criterios relevantes para cumplir la subfunción que tienen asignada.

Tabla 12. Selección del concepto “Sensor pH”.

Criterio para la selección del dispositivo	Importancia (1 - 5)	EPH-M11-Pt100-HT		AP301/21020004	
		Cal	Tot	Cal	Tot
Compatibilidad	5	5	25	5	25
Eficiencia	5	5	25	5	25
Consumo energético	5	5	25	5	25
Costo	5	5	25	5	25
Disponibilidad de adquisición	5	5	25	5	25
Facilidad de implementación	4	4	16	3	12
Garantía	5	5	25	5	25
Total		34	166	33	162
Dispositivo seleccionado		X			

5.1.4.4 Evaluación del concepto “Sensor temperatura”. Debido a que los sensores de pH evaluados anteriormente tienen integrado un sensor pt100 para la compensación de temperatura. De acuerdo eso el electrodo de pH elegido será también el encargado de sensar la temperatura.

5.1.4.5 Evaluación del concepto “Instrumentación de la señal de pH”

Información Técnica

Transmisor DMM4000-pH

Figura 43. Trasmisor HT-DMM4000-pH.



Fuente: Design. Transmisor DMM4000-pH. [En línea] [Citado el: 10 de 01 de 2010.] <http://www.desin.com/principal.asp>.

Fabricante:	Desin Instruments
Rango:	0.00 - 14.00 pH
Precisión:	±0.014 pH
Impedancia:	1000 GΩ
Voltaje de alimentación:	85 - 265 VAC
Formato de salida pH:	4 - 20 mA
Formato de salida temperatura:	4 - 20 mA
Display local:	Si
Salidas de relé:	2
ATC:	Si
Costo:	\$ 688.000 COP

Figura 44. Trasmisor PHTX-271-1P.



Fuente: Omega. [En línea]
[Citado el: 10 de 01 de 2010.]
<http://www.omega.com/Green/pdf/PHTX271.pdf>.

Información Técnica

Transmisor PHTX-271-1P

Fabricante:	Omega
Rango:	0.00 - 14.00 pH
Precisión:	±0.03 pH
Impedancia de entrada:	1000 GΩ
Voltaje de alimentación:	12/24 VDC
Formato de salida pH:	4 - 20 mA
Formato de salida temperatura:	4 - 20 mA
Display local:	Si
Salidas de relé:	1
ATC:	Si
Costo:	\$ 1.046.500 COP

Los criterios para la evaluación de este concepto así como su importancia relativa en la evaluación se muestran en la Tabla 13.

Tabla 13. Criterios y nivel de importancia para la selección del concepto “Instrumentación de la señal de pH”.

Criterio	Nivel de importancia
Compatibilidad	5
Eficiencia	5
Costo	5
Disponibilidad de adquisición	5
Facilidad de implementación	5
Garantía	5

En la Tabla 14 se encuentran los conceptos que van a ser evaluados según los criterios relevantes para cumplir la subfunción que tienen asignada.

Tabla 14. Selección del concepto “Instrumentación de la señal de pH”.

Criterio para la selección del dispositivo	Importancia (1 - 5)	DMM4000-pH		PHTX-271-1P	
		Cal	Tot	Cal	Tot
Compatibilidad	5	5	25	5	25
Eficiencia	5	5	25	5	25
Consumo energético	5	5	25	5	25

Tabla 14. (Continuación).

Criterio para la selección del dispositivo	Importancia (1 - 5)	DMM4000-pH		PHTX-271-1P	
Costo	5	5	25	3	15
Disponibilidad de adquisición	5	5	25	4	20
Facilidad de implementación	4	5	20	3	12
Garantía	5	5	25	3	15
Total		35	170	28	137
Dispositivo seleccionado		X			

5.1.4.6 Evaluación del concepto “Instrumentación de la señal de temperatura”

Debido a que los transmisores de pH evaluados anteriormente tienen la capacidad de transmitir también la señal de temperatura, el transmisor de pH elegido será también el encargado de transmitir la temperatura.

5.1.4.7 Evaluación del concepto “Permitir restringir flujo de solución ácida/básica”

Figura 45. Bomba peristáltica SR25-S300. **Información Técnica**



Fuente: Thomas Industries. [En línea] [Citado el: 10 de 01 de 2010.] <http://www.gd-thomas.com/workareashowcontent.aspx?id=22882>.

Bomba peristáltica SR25-S300 20252201

Fabricante:	Thomas Industries
Tipo de motor:	Motor de pasos
mililitros por revolución:	0.7 ml/rev
Voltaje de alimentación:	24 VDC
Costo:	\$ 592.000 COP

Figura 46. Bomba peristáltica M045 D.C. **Información Técnica**



Fuente: Verder Flex. [En línea] [Citado el: 10 de 01 de 2010.] <http://www.verderflex.com/file.php/281/M045%5B1%5D.pdf>.

Bomba peristáltica M045 D.C.

Fabricante:	Verder Flex
Tipo de motor:	Motor DC
mililitros por revolución:	0.14 ml/rev
Voltaje de alimentación:	12/24 VDC
Costo:	\$ 300.000 COP

Los criterios para la evaluación de este concepto así como su importancia relativa en la evaluación se muestran en la Tabla 15.

Tabla 15. Criterios y nivel de importancia para la selección del concepto “Instrumentación de la señal de pH”.

Criterio	Nivel de importancia
Compatibilidad	5
Eficiencia	5
Costo	5
Disponibilidad de adquisición	4
Facilidad de implementación	5
Garantía	5
Mantenimiento	4
Precisión	5

En la Tabla 16 se encuentran los conceptos que van a ser evaluados según los criterios relevantes para cumplir la subfunción que tienen asignada.

Tabla 16. Selección del concepto “Actuadores de pH”.

Criterio para la selección del dispositivo	Importancia (1 - 5)	SR25-S300 20252201		M045 D.C.	
		Cal	Tot	Cal	Tot
Compatibilidad	5	4	20	5	25
Eficiencia	5	5	25	4	20
Costo	4	2	8	4	16
Disponibilidad de adquisición	5	5	25	5	25
Fácil implementación	4	4	16	5	20
Garantía	5	5	25	4	20
Mantenimiento	5	4	20	4	20
Precisión	5	5	25	3	15
Total		34	164	34	161
Dispositivo seleccionado		X			

5.1.4.8 Evaluación del concepto “Etapa de potencia de los actuadores pH”

Los criterios para la evaluación de este concepto así como su importancia relativa en la evaluación se muestran en la Tabla 17.

Tabla 17. Criterios y nivel de importancia para la selección del concepto “Etapa de potencia de los actuadores de pH”.

Criterio	Nivel de importancia
Seguridad	5
Eficiencia	4
Costo	5
Disponibilidad de adquisición	5
Fácil implementación	5
Garantía	5
Mantenimiento	5

En la Tabla 18 se encuentran los conceptos que van a ser evaluados según los criterios relevantes para cumplir la subfunción que tienen asignada.

Tabla 18. Selección del concepto “Etapa de potencia de los actuadores de pH”.

Criterio para la selección del dispositivo	Importancia (1 - 5)	L297		Circuito con transistores de potencia	
		Cal	Tot	Cal	Tot
Seguridad	5	4	20	5	25
Eficiencia	4	5	20	5	20
Costo	5	5	25	4	20
Disponibilidad de adquisición	5	5	25	5	25
Fácil implementación	5	5	25	4	20
Garantía	5	5	25	5	25
Mantenimiento	5	4	20	4	20
Total		33	160	32	155
Dispositivo seleccionado		X			

5.1.4.9 Evaluación del concepto “Permitir/Restringir el paso de líquido refrigerante”

Figura 47. Válvula solenoide 2W040.



Válvula Solenoide 2W040

Fabricante:	STC
Diámetro:	3/8.
Cableado:	2 hilos
Temperatura de operación:	-5 – 80 °C
Presión de operación máxima:	100 PSI
Costo:	\$ 68.000 COP

Figura 48. Válvula solenoide EV210A.



Información Técnica EV210A

Fabricante:	Danfoss
Diámetro:	1/4.
Cableado:	2 hilos
Temperatura de operación:	-30 – 120 °C
Presión de operación máxima:	725 PSIG
Costo:	\$ 128.000 COP

5.1.5 Establecer especificaciones finales. Después de realizar la selección de cada uno de los dispositivos para los diferentes conceptos que resultaron de la descomposición funcional del sistema de control del biorreactor se presentan las especificaciones técnicas que con la integración de estos dispositivos tendrá el sistema.

Especificaciones técnicas finales:

- Disponibilidad de entradas analógicas: 13.
- Disponibilidad de E/S digitales: 36
- Dispositivo de control: PIC18f4520
- Tipo de controlador: control PID difuso.
- Amigabilidad de la Interfaz: Cómoda.
- Rango de pH: 0 – 14.
- Rango de Temperatura: 0 °C – 115 °C.
- Precisión del sensor de pH: ± 0.014
- Precisión del sensor de Temperatura: $\pm 0.1^{\circ}\text{C}$.
- Costo de producción: 2.500.000.
- Precisión del control de pH: ± 0 .
- Precisión del control de Temperatura: $\pm 1^{\circ}\text{C}$.

- Tipo de actuador de pH: Bomba peristáltica SR25-S300 20252201.
- Tipo de actuador temperatura: Resistencia tipo abrazadera.

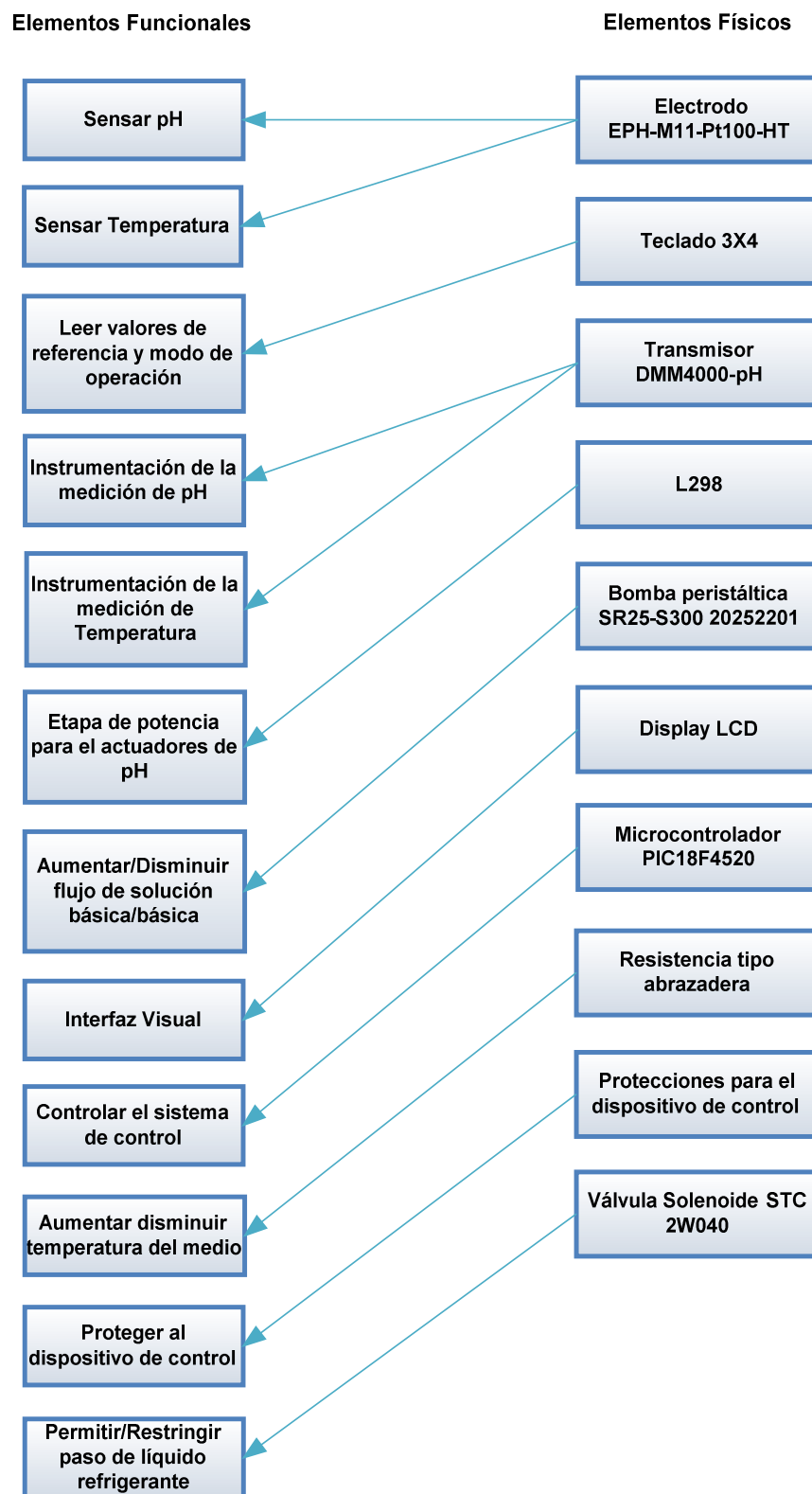
5.2 Diseño a nivel de sistemas

En esta etapa se establece la arquitectura del producto mediante la descomposición del sistema control del biorreactor y el análisis de las interacciones existentes entre estos. A su vez se identifican los diferentes canales de compra y distribuidores para cada uno de los dispositivos que resultaron de la selección de conceptos.

5.2.1 Estudio y selección de la arquitectura. Para el desarrollo del sistema de control del biorreactor, se va a utilizar el concepto de arquitectura modular según el cuál el producto se desarrolla en diferentes módulos que individualmente se encargan de cumplir al menos una subfunción asignada de la descomposición funcional.

En la Figura 49 se muestra que módulo funcional cumple cada uno de los elementos físicos que han sido seleccionados para formar parte del sistema de control.

Figura 49. Relación entre elementos funcionales y elementos físicos del sistema de control.



5.3 Diseño detallado

En esta etapa se describirá con detalle como opera cada una de los módulos que componen el sistema de control, así como también se mostrará el código con el que fue programado el microcontrolador.

5.3.1 Diseño para manufactura (DPM). Cuando se habla de DPM, se está especificando el diseño a nivel del sistema, con el fin de valorar la implementación modular que tiene el producto, manejando la reducción de los costos de los elementos, así como la fácil consecución de los mismos y disminuir la complejidad para el proceso de manufactura.

Pensando en la manufactura, se diseño una caja en la que irán ubicados todos los dispositivos involucrados con el sistema de control, a excepción de la resistencia, el intercambiador y la válvula del intercambiador por obvias razones.

Se integraron piezas estándares para la elaboración del sistema, así mismo componentes electrónicos de fácil ubicación en el mercado, para de esta manera minimizar los costos de fabricación. Con esto se mejoro la calidad del sistema ya que la funcionalidad obtendrá mejor respuesta por ser componentes electrónicos ya integrados teniendo así menores interferencias y fallas.

5.3.1.1 Costo de componentes y Materiales. Dentro de los componentes utilizados se identificaron dos clases, las partes estándar, las cuales están constituidas por componentes presentes en el comercio local, lo cual hace de ellos un punto importante en el desarrollo del sistema ya que el hecho de utilizarlos agiliza tanto la construcción como también la reparación ante cualquier daño, y la característica principal es que están presentes en el mercado lo cual es un factor favorable ya que su presencia y fácil consecución disminuye los costos del dispositivo; por otro lado se encuentran las partes propias, estas son construidas a partir de materia prima y constituyen la carcasa y estructura donde está ubicado el sistema de control.

Partes Estándar: Se compran componentes electrónicos como: sensores, pulsadores, resistencias, condensadores, integrados, cristales de reloj etc. componentes que forman parte del sistema actuador como: resistencias bombas peristálticas y algunas partes mecánicas como tortillería en general, arandelas chazos, remaches etc.

Partes Propias: Fabricadas según los requerimientos del diseño a partir de la materia prima, en esta parte se destacan materiales como lamina de hierro, laminas de aluminio, demás elementos utilizador para la construcción de la estructura dispositivo y carcasa del sistema de control.

5.3.1.2 Lista de componentes

Tabla 19. Lista general de componentes.

Costos sistema actuador			
Componente	Cantidad	Precio Unit.	Valor Total
Válvula para el intercambiador	1	68.000	60.000
Resistencia de potencia	1	48.000	10.000
Bombas peristálticas	2	592.000	1.184.000

Tabla 19. (Continuación).

Componente	Cantidad	Precio Unit.	Valor Total
Carcasa protectora	1	15.000	15.000
Tornillería en general			25.000
TOTAL			1.294.000

Tabla 20. Lista general de componentes.

Costos sistema electrónico			
Componente	Cantidad	Precio Unit.	Valor Total
1N4001	9	\$ 100	\$ 900
1N914	8	\$ 200	\$ 1.600
2N2222	1	\$ 250	\$ 250
LM7805	1	\$ 1.200	\$ 1.200
Capacitor 0,33uf	1	\$ 300	\$ 300
Capacitor 0,1uf	1	\$ 400	\$ 400
capacitor 1000uf 25V	1	\$ 50	\$ 50
capacitor 470uf 25V	1	\$ 300	\$ 300
Resistencias	20	\$ 50	\$ 1.000
Respack 7	1	\$ 200	\$ 200
Fusible 2A	1	\$ 300	\$ 300
Teclado matricial	1	\$ 10.000	\$ 10.000
L298	2	\$ 13.800	\$ 27.600
Display 2x16	1	\$ 16.000	\$ 16.000
LM358	1	\$ 800	\$ 800
MOC3021	1	\$ 1.500	\$ 1.500
4N25	1	\$ 1.300	\$ 1.300
PIC18f4520	1	\$ 20.000	\$ 20.000
Varistor 120V	1	\$ 2.000	\$ 2.000
Transformador 20V-0V-20V 1,5A	1	\$ 45.000	\$ 45.000
Trasformador 9V-0V-9V y 6V-0V-6V 1A	1	\$ 15.000	\$ 15.000
Conector bornera	4	\$ 300	\$ 1.200
Transmisor DMM4000-pH	1	\$ 688.000	\$ 688.000
Electrodo EPH-M11-Pt100-HT	1	\$ 362.000	\$ 362.000
TOTAL			\$ 1.196.900

Finalmente para estimar los costos totales del sistema se realiza la suma de los costos obtenidos anteriormente.

Tabla 21. Costo total del dispositivo.

Costos del sistema en general	
Descripción	Valor Total
Valor del sistema actuador	\$ 1.294.000
Valor del sistema electrónico	\$ 1.196.900
VALOR TOTAL PARA EL DESARROLLO DEL DISPOSITIVO	\$ 2.490.900

Mediante el análisis de manufactura, se pueden observar ciertas limitantes en el desarrollo de algunos elementos que conforman el sistema, pues estos según su geometría y funcionamiento deben fabricarse mediante diferentes procesos tales como maquinado, pulido, inyección, etc. Sin embargo se pretende que la mayoría de componentes sean de fácil consecución en el mercado con el fin de brindar versatilidad al diseño.

5.3.2 Diseño para mantenimiento (DPMT). El plan de mantenimiento está determinado en la Tabla 22.

Tabla 22. Acciones de mantenimiento.

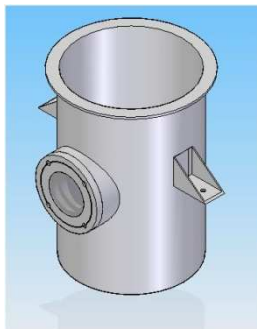
PARTES	DESCRIPCION DEL MANTENIMIENTO	REVISAR (DIAS)	ACCION DE MANTENIMIENTO
Válvula del intercambiador.	Revisar voltaje, corriente, conexiones.	120	Cambiar conexiones
Bombas peristálticas.	Realizar mantenimiento preventivo de motores.	30	Limpiar, revisar funcionamiento.
	Revisar la condición de la manguera externa.	30	Limpiar, cambiar si es necesario.
Plaqueta electrónica	Revisar voltajes, corriente y potencia en los puntos de prueba.	60	Cambiar dispositivo defectuoso
Sensores	Revisar voltajes de salida	45	Calibrar los sensores, cambiar el dispositivo defectuoso
	Revisar alimentación	45	
Transmisor	Revisar voltajes de salida	45	Calibrar los sensores, cambiar el dispositivo defectuoso

5.3.3 Diseño Físico del dispositivo automatizado

El dispositivo final comprende todos los elementos que existían previamente así como todos los elementos que forman parte del sistema de automatización.

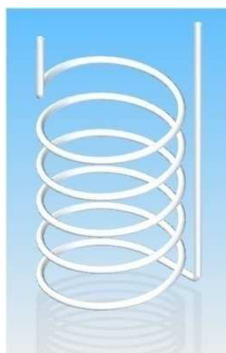
En primera medida podemos ver el recipiente hecho de acero inoxidable en la Figura 50. Los planos del recipiente se encuentran en el Anexo 5.

Figura 50. Recipiente del biorreactor.



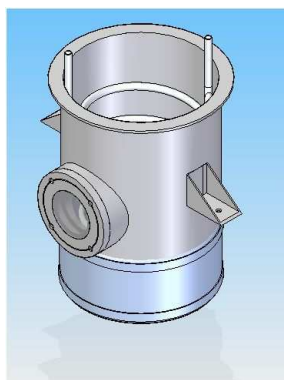
El intercambiador de calor se ubica en el interior del recipiente; está hecho de tubing de acero inoxidable con diámetro de 3/8. El intercambiador se encuentra en la Figura 51.

Figura 51. Intercambiador de calor



La resistencia en forma de abrazadera encargada del calentamiento de la muestra se ubica rodeando la parte inferior del recipiente, tal como se muestra en la Figura 52.

Figura 52. Ensemble del intercambiador la resistencia y el recipiente.

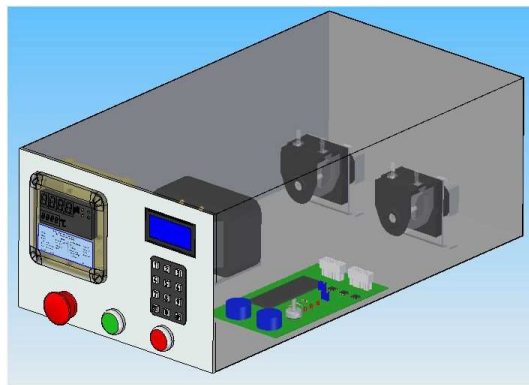


En la caja del dispositivo de control están ubicados los siguientes elementos:

- La interfaz hombre máquina (display LCD, teclado matricial y pulsadores).
- Las bombas peristálticas.
- La plaqueta electrónica.
- El transformador.
- El trasmisor.

El dispositivo de control se ilustra en la Figura 53.

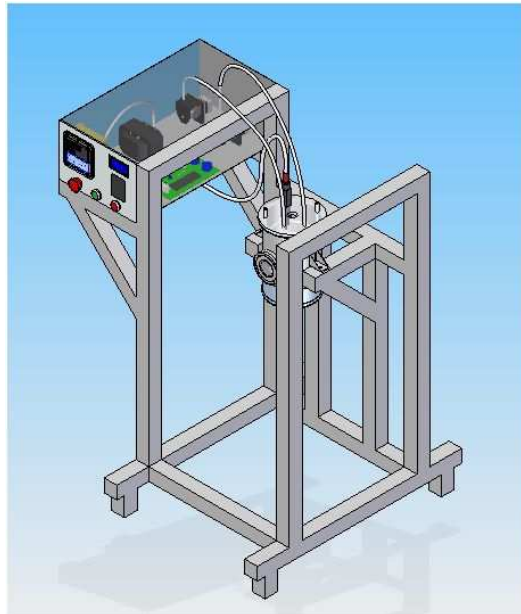
Figura 53. Dispositivo de control.



Las dimensiones de la carcasa se encuentran en el Anexo 6.

En la Figura 54 se muestra la estructura en donde van ubicados el recipiente y el dispositivo de control del biorreactor.

Figura 54. Biorreactor con el sistema de automatización.



5.3.4 Diseño del circuito electrónico. El circuito electrónico que se implementó se encuentra en el

5.3.5 Anexo 1 en el que se pueden destacar los siguientes bloques:

- Fuente de 5V DC.
- Fuente de 24V DC.
- Detección de cruce por cero.
- Etapa de potencia para la resistencia.
- Etapa de potencia para la válvula del intercambiador de calor.
- Etapa de potencia para los motores.
- Interfaz hombre-máquina.
- Sensores.

A continuación se va a explicar detalladamente cada uno de los bloques anteriormente mencionados.

5.3.5.1 Fuente de 5V DC

Al principio de la fuente encontramos un transformador con tap central, del cual sólo se utilizó un terminal de salida del bobinado secundario y el tap central. A la señal de salida del terminal se le realiza una rectificación de onda completa con los diodos D1 y D2 de referencia 1N4001. Después de realizar la rectificación la señal pasa por un filtro que se encarga de reducir el rizado de la señal rectificada, esto para ser enviada al regulador de voltaje LM7805. Los condensadores cerámicos en la entrada y la salida del condensador son sugeridos por el fabricante del regulador para tener un mejor desempeño del mismo. Por último, la señal pasa de nuevo por un filtro con el fin de asegurar la estabilidad de la misma.

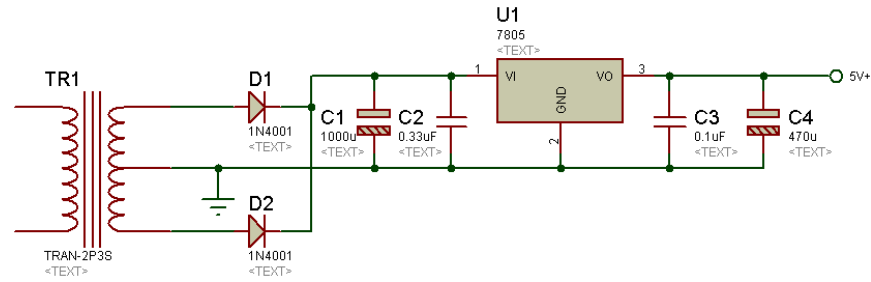
El criterio de selección del transformador se explica a continuación:

Dado que deseamos obtener 5V a la salida de la fuente debemos garantizar que el voltaje rectificado sea mayor a 5V, de acuerdo a esto hacemos el cálculo para saber el voltaje a la salida del transformador:

$$V_{DC} < \frac{V_{PP}}{\pi}$$
$$V_{DC} < \frac{2 \cdot V_{RMS} \cdot \sqrt{2}}{\pi}$$
$$\frac{V_{DC} \cdot \pi}{2 \cdot \sqrt{2}} < V_{RMS}$$
$$\frac{5 \cdot \pi}{2 \cdot \sqrt{2}} \approx 5.55 < V_{RMS}$$

El transformador que se utilizó tiene un voltaje de 9V en el bobinado secundario.

Figura 55. Fuente de 5V DC.



5.3.5.2 Fuente de 24V DC

La fuente de 24V DC tiene la misma estructura que la 5V DC, sin embargo el condensador de 1000μF se cambio por uno de 2200μF y el transformador debe tener la capacidad de entregar 27V RMS.

5.3.5.3 Detección de cruce por cero

Antes de realizar la detección de cruce por cero debemos cambiar el formato de la señal que proviene del transformador y adecuarlo para poder enviar la señal al microcontrolador. Esto lo conseguimos con un amplificador sumador no inversor de la siguiente forma:

Si el transformador utilizado para este circuito entrega 6V RMS tenemos que:

$$V_p = V_{RMS} \cdot \sqrt{2} = 6 \cdot \sqrt{2} \approx 8.49$$

Dado que la señal tiene un voltaje pico más alto del que permite el PIC (5V) le daremos a esta señal una ganancia de $\frac{1}{4}$ en el sumador para así obtener un voltaje pico de 2.15V.

La ecuación de un sumador inversor como el que se muestra en la Figura 56, está dada por la fórmula:

Ecuación 6. Voltaje de salida de un sumador no inversor.

$$V_o = \frac{R_3}{R_1} V_1 + \frac{R_3}{R_2} V_2$$

Siempre y cuando se cumpla la siguiente condición:

$$\frac{R_1 \cdot R_2}{R_1 + R_2} = \frac{R_3 \cdot R_4}{R_3 + R_4}$$

Para éste caso específico V2 es la señal que proviene del transformador. Dado que la intención es atenuar la señal dándole una ganancia de $\frac{1}{4}$ tenemos que:

$$\frac{R3}{R2} = \frac{1}{4}$$

Si asignamos el valor de $R3=1K$, entonces $R2=4K$.

Hasta ahora nos hemos asegurado de que el valor pico de la señal de salida es de $2.15V$, sin embargo la señal aún tiene una parte negativa, razón que no nos permite llevar la señal al microcontrolador.

La solución a este problema radica en sumar una componente de DC tal que el valor mínimo de la señal de salida sea cero. Por esta razón el valor de la señal de DC es de aproximadamente $2.1V$, no obstante debemos asegurarnos que la ganancia del sumador a esta señal sea igual a la unidad. Dicho esto se tiene que:

$$\frac{R3}{R1} = 1$$

Dado que ya se había asignado que $R3=1K$, entonces $R1=1K$.

Dado que debemos garantizar que:

$$\frac{R1 \cdot R2}{R1 + R2} = \frac{R3 \cdot R4}{R3 + R4}$$

Entonces reemplazamos:

$$\frac{1k \cdot 4k}{1k + 4k} = \frac{1k \cdot R4}{1k + R4}$$

A simple vista se puede ver que $R4=4K$.

Los $2V$ de la componente DC que se desea adicionar los obtenemos de un divisor de tensión aplicado a la fuente DC de $5V$, los cálculos se muestran a continuación:

$$\frac{R6}{R6 + R8} \cdot 5V = 2.1V$$

Si $R6+R8=1K$, entonces:

$$\frac{R6}{1K} \cdot 5V = 2.1V$$

Resolviendo obtenemos que $R6=420\Omega$, entonces $R8=580\Omega$

Figura 56. Sumador no inversor.

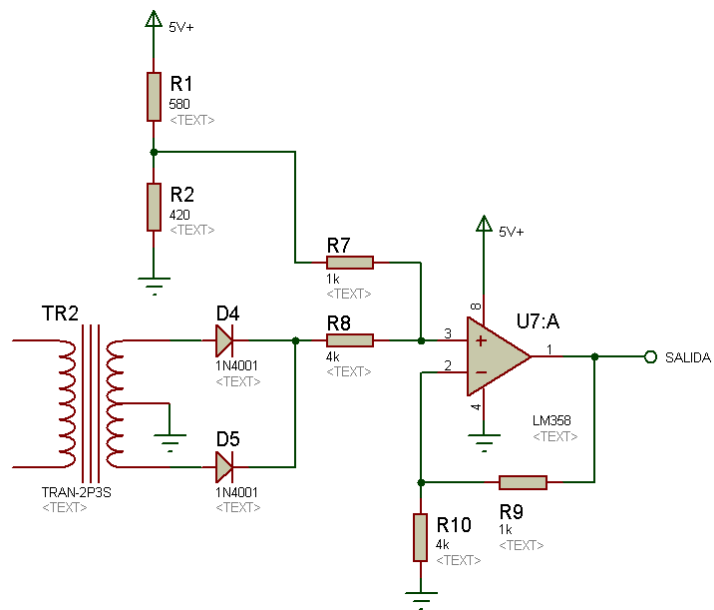
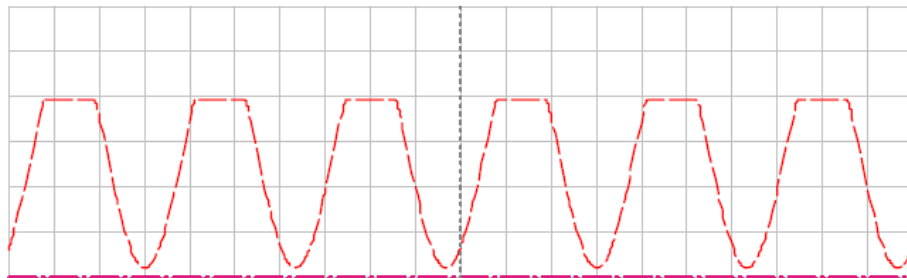


Figura 57. Señal de salida del sumador no inversor.



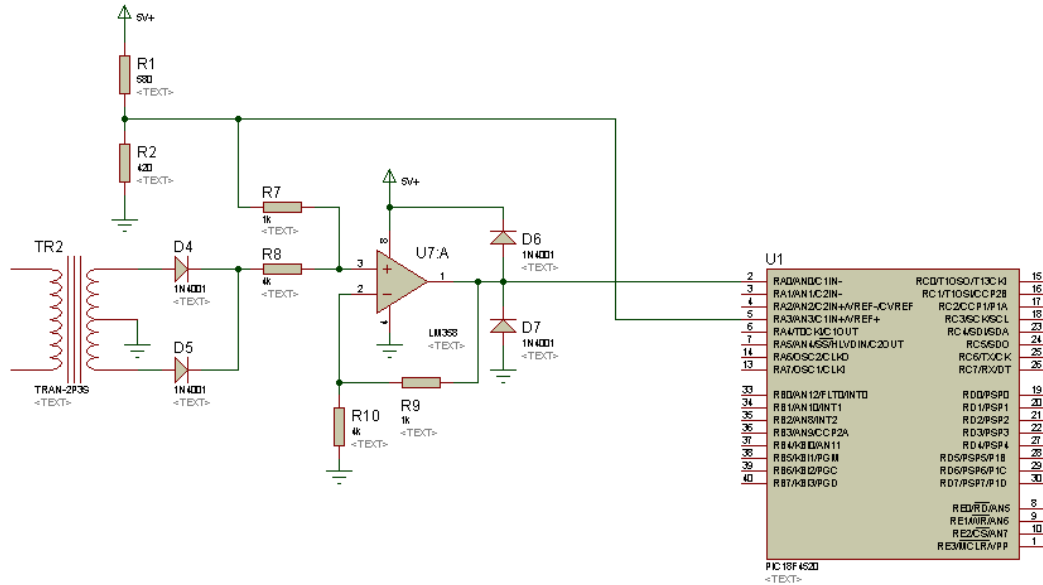
En la Figura 57 la señal roja representa la salida del sumador no inversor, y la señal lila indica tierra. La escala vertical es de un voltio por división.

La señal obtenida en la salida del sumador no inversor es la que se va a comparar para realizar la detección de cruce por cero en el microcontrolador. La señal de referencia es la misma que se utilizó para adicionar la componente DC en el sumador.

El esquema de conexión entre esta etapa y el PIC se muestra en la

Figura 58.

Figura 58. Esquema de conexión al microcontrolador del bloque de detección de cruce por cero.



5.3.5.4 Etapa de potencia para la resistencia

La regulación de voltaje de la resistencia se hace por medio del triac, sin embargo debemos opto aislar la parte de potencia del PIC, y esto lo logramos con el opto-triac MOC2022.

Según los datos del fabricante la corriente del LED necesaria para asegurar el disparo es de 10 mA y la caída de tensión en el LED es de 1.2V. De acuerdo a debemos calcular la resistencia a la entrada del diodo LED.

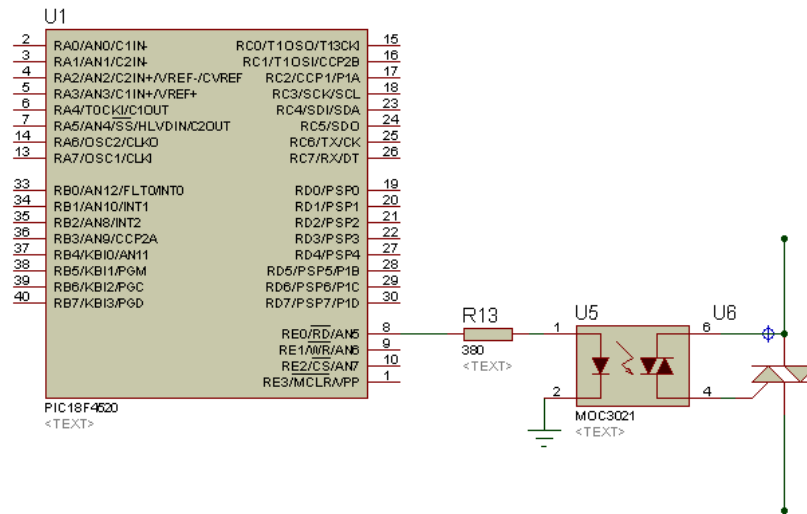
$$V_{LED} = I_{LED} \cdot R$$

$$\frac{V_{LED}}{I_{LED}} = R$$

$$\frac{5V - 1.2V}{10 \text{ mA}} = R$$

De ahí obtenemos que $R=380\Omega$

Figura 59. Etapa de potencia para la resistencia.



La Figura 59 muestra la conexión entre el micro controlador y el triac.

5.3.5.5 Etapa de potencia para la válvula del intercambiador de calor

La válvula del intercambiador de calor está gobernada con un pin del microcontrolador con lógica negativa, lo que quiere decir que cuando en el pin del microcontrolador está en un 0 lógico está abierta la válvula y cuando el pin está en un 1 lógico la válvula esta cerrada.

La etapa de potencia para la válvula consiste en un transistor 2N2222, sin embargo es recomendable opto-aislar la salida del microcontrolador, por lo que se implementó implementando un pre-driver con un opto-transistor 4N25.

Para que el 4N25 se active el fabricante nos dice que debemos garantizar una corriente en el diodo emisor de 50mA y una corriente de Colector de 2mA, entonces:

$$R_{LED} = \frac{V_{LED}}{I_{LED}}$$

$$R_{LED} = \frac{5V - 1.2V}{50 \text{ mA}} = 76\Omega$$

Ahora se debe hallar un Rc que permita tener una corriente de colector de 2mA:

$$R_C = \frac{V_C}{I_C}$$

$$R_C = \frac{24V}{2\text{mA}} = 12K\Omega$$

La salida del opto-transistor se envía a la base del transistor 2N2222, el cual tendrá una configuración emisor común y al que además adecuaremos para trabajar en corte y saturación.

Se designo un valor de $R_c=1k$, a partir de ahí se tiene que:

$$I_{C\ sat} = \frac{V_{CC}}{R_c}$$

$$I_{C\ sat} = \frac{24V}{1K\Omega} = 24mA$$

Este es un valor aceptable de I_c ya que es menor que el I_c máximo que da el fabricante (800mA).

El valor de β que se utilizó corresponde a 100, que es el valor mínimo con el que trabaja el transistor según el fabricante. Con estos valores se debe hallar la corriente de base de la siguiente forma:

$$I_b = \frac{I_{C\ sat}}{\beta}$$

$$I_b = \frac{24mA}{100} = 0.24mA$$

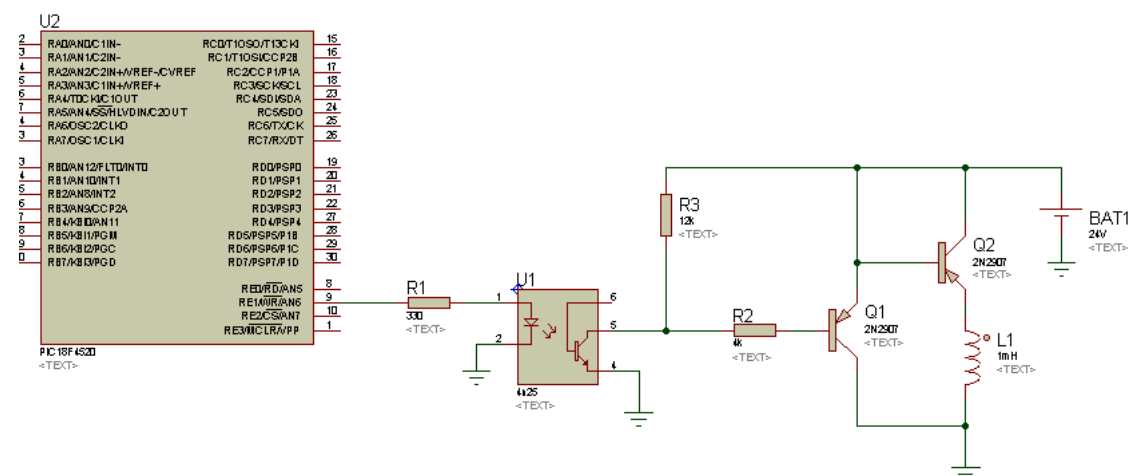
Ahora se que halla R_b :

$$R_b = \frac{V_b}{I_b}$$

$$R_b = \frac{5V}{0.24mA} \approx 21K\Omega$$

El esquema para la conexión del microcontrolador a la válvula del intercambiador se muestra en la Figura 60.

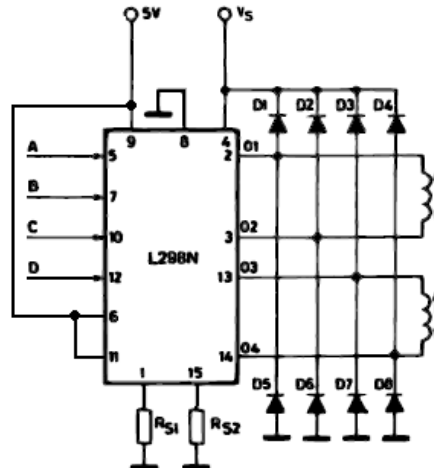
Figura 60. Etapa de potencia para la válvula del intercambiador de calor.



5.3.5.6 Etapa de potencia para los motores

La etapa de potencia de cada motor se implemento con driver L298 siguiendo un esquema planteado por el fabricante.

Figura 61. Esquema de la implementación del L298.

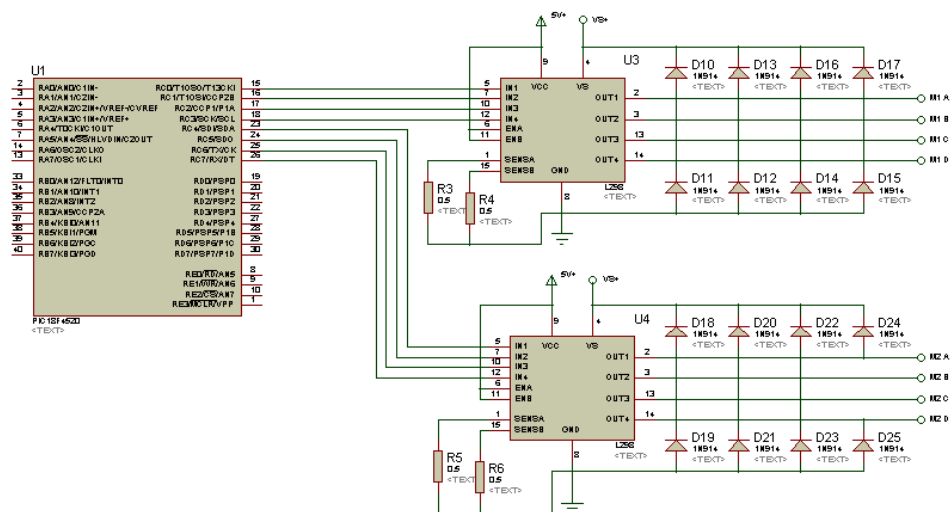


Los diodos mostrados en la Figura 61 son diodos de conmutación rápida, y para la implementación se utilizaron diodos 1N914.

Las resistencias S1 y S2 conectadas a los pines 1 y 15 permiten sensor la corriente consumida por los motores y el fabricante recomienda que sea de 0.5Ω . Además de esto el fabricante asegura este esquema permite manejar motores que consuman hasta 2A.

La conexión del microcontrolador al L298 puede hacerse directamente como se muestra en la Figura 62.

Figura 62. Esquema de conexión entre el PIC18f4520 y el L298.



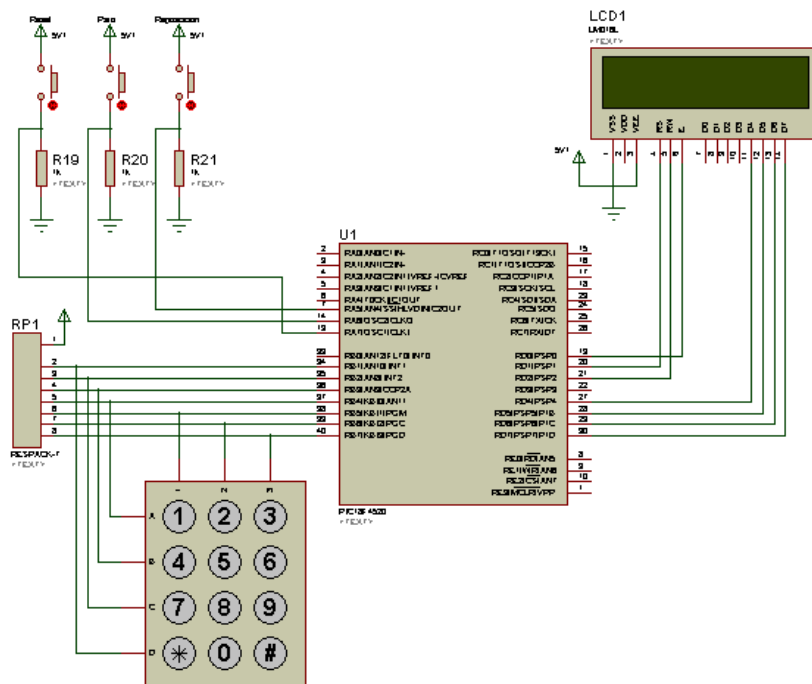
5.3.5.7 Interfaz hombre máquina

La interfaz del dispositivo consiste en un display lcd de 2x16, un teclado matricial y tres pulsadores.

El display tiene una conexión directa con el microcontrolador mientras que el teclado requiere de la conexión de resistencias pull-up para su correcto funcionamiento.

El esquema de la interfaz se muestra en la Figura 63.

Figura 63. Esquema de la interfaz hombre máquina.



5.3.5.8 Sensores. Las señales provenientes del transmisor tienen un formato de 4-20mA así que cada señal debe pasar una resistencia Shunt con el fin de obtener la medición de la señal en términos de voltaje. Con un valor de la resistencia de 250Ω obtenemos un rango de 1-5V, por lo que será necesario conectar el pin de referencia inferior del ADC a 1V.

5.3.6 Diseño del Software. El código fue diseñado para realizar el control de las variables pH y temperatura, por ende esta sección se divide en dos partes importantes: el control de pH y el control de temperatura.

5.3.6.1 Control de temperatura. El control de temperatura que se implementó en el dispositivo consta básicamente de los siguientes bloques:

- Controlador difuso para el cálculo del voltaje de control de la resistencia.

- Detección de cruce por cero.
- Regulación de voltaje por ángulo de fase.
- Control On – Off para el cálculo del voltaje de control del intercambiador.

El control de temperatura que a continuación se describe de manera detallada tiene como objetivo calcular el valor que para cada instante de tiempo t deberán recibir la resistencia y la válvula que permite o restringe el paso de agua al intercambiador de calor; pero debido a que la resistencia trabaja en corriente alterna, la regulación se hace por ángulo de disparo, por ende es necesario realizar una detección de cruce por cero.

A continuación se va a explicar con detalle cada uno de los bloques mencionados.

Detección del cruce por cero

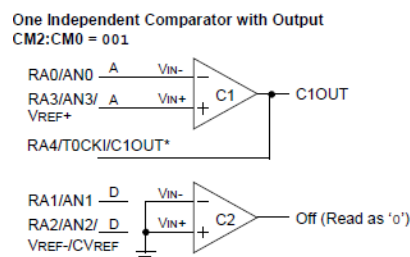
Dado que el PIC18f4520 cuenta con un comparador analógico programable, la detección de cruce por cero se realizó por software usando este módulo del PIC y su interrupción asociada. Para la utilización de dicho método debe realizarse la configuración del comparador dentro del main() así como la activación de la interrupción asociada al comparador y la interrupción global.

```
void main()
{
    enable_interrupts(INT_COMP);
    enable_interrupts(GLOBAL);

    CMCON= 0b00000001;
}
```

Según el valor cargado al registro CMCON el módulo comparador es configurado para trabajar con un solo comparador con un pin de salida asociado, esto con el fin de evitar el uso del pin A2 que se necesita en la configuración del adc. En la Figura 64 se muestra la configuración del comparador correspondiente al valor 01H cargado en el registro CMCON

Figura 64. Configuración del módulo comparador correspondiente al valor 01H cargado en el registro CMCON.



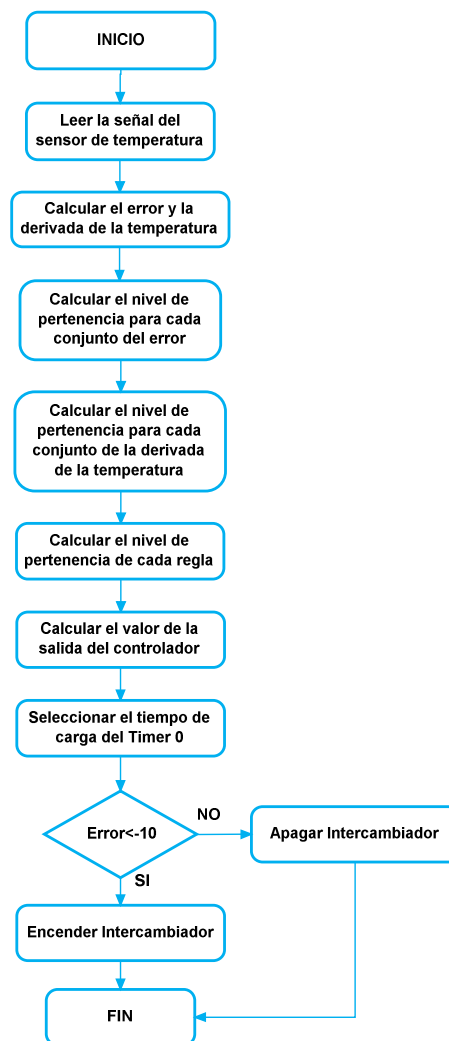
El código de la rutina de interrupción se muestra a continuación:

```
#int_COMP
void COMP_isr()
{
    if(C1OUT){
        output_toggle(PIN_D0);
        delay_us(10);
        output_toggle(PIN_D0);
    }
```


El código de la simulación presentada difiere del código que se usa para el programa de la implementación. La única diferencia que tienen los códigos (en cuanto a la detección de cruces por cero se refiere) es que el código de la simulación tiene como fin generar un pulso cuando se produce el cruce por cero mientras el código de la implementación carga un valor en el Timer 0.

La secuencia que realiza la tarea Temp, encargada de realizar el control de temperatura, está descrita en la Figura 67.

Figura 67. Diagrama de flujo de la tarea Temp.



Las variables globales que utiliza la tarea son:

```

//CONSTANTES GLOBALES DEL CONTROLADOR DE TEMPERATURA
int ref;
int16 control;
float valor,x;
float dtemp = 0.0;
float error = 0.0;
  
```

```

float valor_1 = 0.0;
float k = 1;
float kd = 5;
float ko = 1.35;
float PertenenciaError[5];
float PertenenciaDTemp[5];
const float SalidaResistencia[5][5]= {{0, 0, 0, 900, 900},
                                         {0, 0, 0, 900, 900},
                                         {0, 0, 0, 900, 900},
                                         {0, 0, 0, 450, 900},
                                         {0, 0, 0, 225, 900}};

```

El código comentado de la tarea, se muestra a continuación:

```

//TAREA CONTROLADOR DE TEMPERATURA
void temp ( )
{
    set_adc_channel(2); //Selecciona el canal del sensor de temp
    valor = read_adc (); //Lee la señal del sensor
    valor = 1 + valor * 4 / 1024.0; //Se normaliza el la señal medida
    error = k * (ref - valor); //Calcula el error * k
    dtemp = kd * (valor - valor_1); //Calcula dtemp * kd
    CalcularPertenencia(error,dtemp); //Calcula nivel de pertenencia
    x = defuzzResistencia(); //Calcula el valor de la salida
    x *= ko; //Salida * ko
    control = x; //Guarda el valor entero de x en control
    if (control > 110) control = 110; //Satura el controlador
    Angulo_Disparo(control); //Asigna el valor al timer0
    valor_1 = valor; //Guarda la medición actual en valor_1
    if(ref-valor < -10)
        output_high(PIN_E1); //Decide si encender el intercambiador
    else output_low(PIN_E1); //Decide si apagar el intercambiador
    rtos_yield(); //Regresa el control al programador de tareas
}

```

El método `set_adc_channel()` permite seleccionar el canal en el que se encuentra conectado el sensor de temperatura, el número entre paréntesis indica el pin correspondiente del canal asociado con el conversor.

El método `read_adc()` entrega un valor entero de 16 bits según la directiva `#device ADC=` empleada. Dicha directiva trabaja según la tabla:

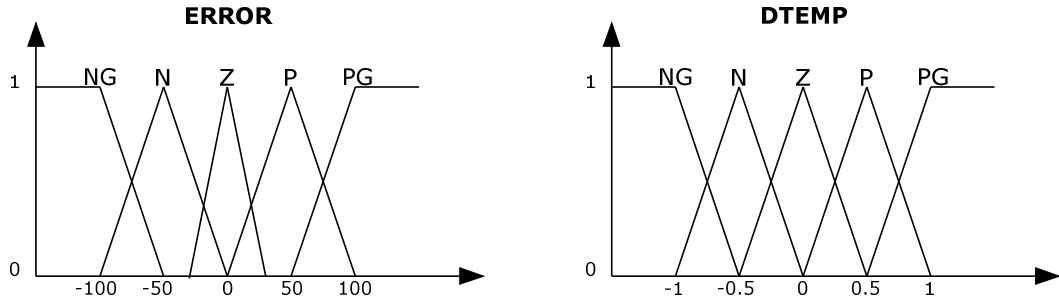
Tabla 23. Rango de valores del ADC según el tipo de dispositivo y la directiva `#device ADC=`.

DEVICE	8 bit	10 bit	11 bit	16 bit
ADC=8	00 - FF	00 - FF	00 - FF	00 – FF
ADC=10	*	0 – 3FF	*	*
ADC=11	*	*	0 - 7FF	*
ADC=16	0 - FF00	0 - FFC0	0 - FFE0	0 - FFFF

El método `Calcular_Pertenencia()` se encarga de calcular las pertenencias del error y la derivada de la temperatura a cada uno de los conjuntos que conforman su universo de discurso respectivamente. Antes de explicar como funciona el método debemos aclarar de qué manera están implementados los conjuntos difusos en el programa.

Para efectos de simplificación en los cálculos los conjuntos difusos sólo tienen funciones de pertenencia de tipo triangular y L, y se definieron así:

Figura 68. Universos de discurso del error y la derivada de la temperatura para el controlador de temperatura.



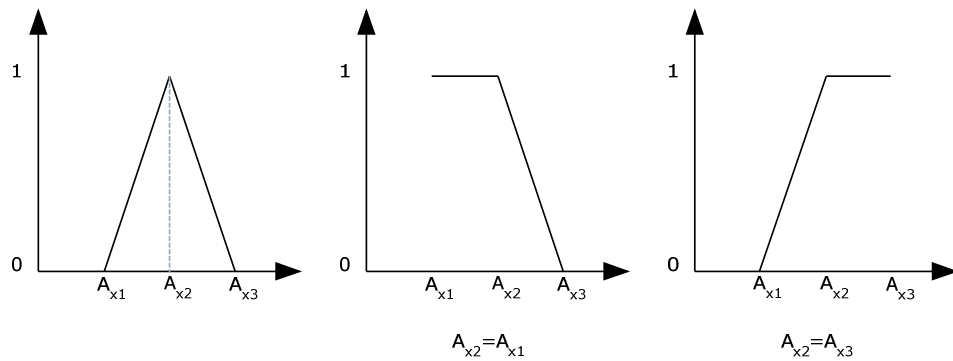
En el programa los conjuntos se encuentran descritos por arreglos de 5x3 definidos así:

Tabla 24. Formato de los arreglos que contienen a los conjuntos difusos del error y la derivada de la temperatura.

A_{11}	A_{12}	A_{12}
A_{21}	A_{22}	A_{22}
A_{31}	A_{32}	A_{32}
A_{41}	A_{42}	A_{42}
A_{51}	A_{52}	A_{52}

Cada fila del arreglo contiene los valores críticos de una variable lingüística. En la Figura 69 se observa la manera en la que están expresados los conjuntos difusos siguiendo la sintaxis de los arreglos.

Figura 69. Representación de los conjuntos difusos según el formato empleado en los arreglos.



El método retorna un arreglo que de 1x5 que contiene las pertenencias de la variable al conjunto de la siguiente manera:

Tabla 25. Formato del arreglo que contiene los valores de pertenencia a los conjuntos difusos.

NG	N	Z	P	PG
----	---	---	---	----

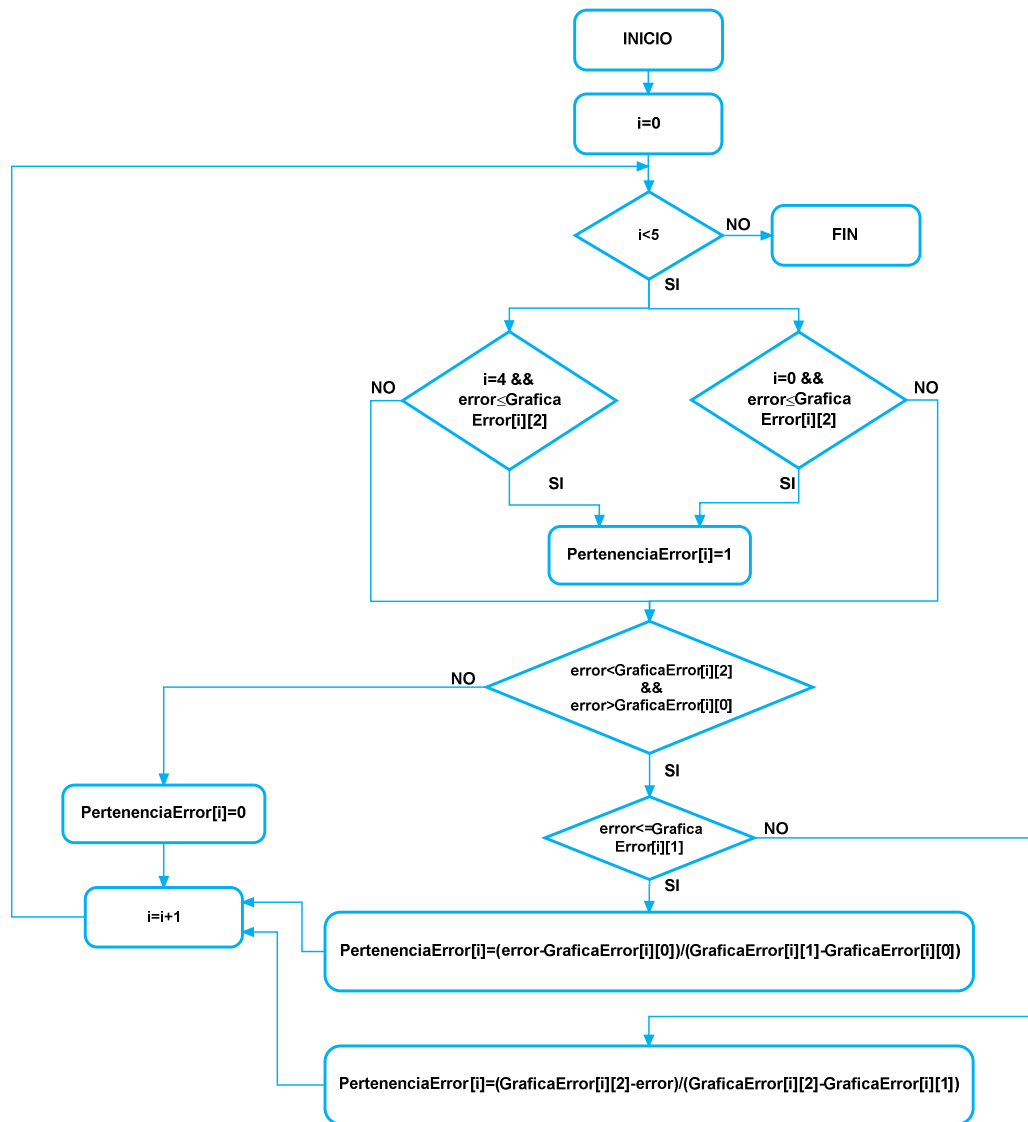
La secuencia que realiza el método Calcular_Pertenencia, esta descrita en la Figura 70.

Como primera instancia el método se encarga de realizar algunas validaciones pertinentes que le permitan entregar una respuesta haciendo el menor número de cálculos posibles.

La primera validación importante que hace es si el valor de la variable es menor o igual al elemento A_{x2} del conjunto NG o mayor o igual a A_{x2} del conjunto PG, pues de ser así el nivel de pertenencia al conjunto es uno.

Después, valida que el valor de la variable se encuentre en el rango (A_{x1}, A_{x3}) de cada conjunto, pues de no ser así el nivel de pertenencia al conjunto es cero. Sin embargo, si el valor de la variable se encuentra en el rango (A_{x1}, A_{x3}) , el controlador deberá validar si este valor esta en el rango $(A_{x1}, A_{x2}]$ o en el rango (A_{x2}, A_{x3}) para así elegir la función con la que determinará el nivel de pertenencia.

Figura 70. Diagrama de flujo del método Calcular_Pertenencia().



La Figura 70 muestra la secuencia que ejecuta el método para calcular el nivel de pertenencia del error a cada conjunto; el cálculo del nivel de pertenencia de la derivada de la temperatura a cada conjunto se hace de la misma forma.

El código comentado se muestra a continuación:

```

void CalcularPertenencia(float error,float dvariable,)
{
    int i;
    //Universo de discurso de la derivada de la temperatura
    const float GraficaDTemp[5][3]= {{ -1 , -1 , -0.5},
                                       { -1 , -0.5 , 0 },
                                       { -0.5 , 0 , 0.5},
                                       { 0 , 0.5 , 1 },
                                       { 0.5 , 1 , 1 }};

    //Universo de discurso del error
    const float GraficaError[5][3]= {{-100 ,-100 , -25},

```

```

        { -50 , -25 , 0 },
        { -25 , 0 , 25 },
        { 0 , 20 , 50 },
        { 25 , 100 , 100 } };

for(i=0;i<5;i++){
    if(i==0&&error<=GraficaError[i][1]||i==4&&error>=GraficaError[i][2]){
        PertenenciaError[i]=1;//Valida si error es <A12 o >A52
    }
    else{
        if(error<GraficaError[i][2]&&error>GraficaError[i][0]){ //Valida si
error
//está en el rango (Ax1,Ax3)
        if(error<=GraficaError[i][1]){ } //Valida si error está en el rango
// (Ax1,Ax2)
            PertenenciaError[i]=(error-GraficaError[i][0])/
            (GraficaError[i][1]-GraficaError[i][0]);
        }else{ } //Valida si error está en el rango (Ax2,Ax3)
            PertenenciaError[i]=(GraficaError[i][2]-
            error)/(GraficaError[i][2]-GraficaError[i][1]);
        }
    }else{ //Si no está en el rango (Ax1,Ax3) entonces el nivel de
pertenencia
        //es 0
        PertenenciaError[i]=0;
    }
}

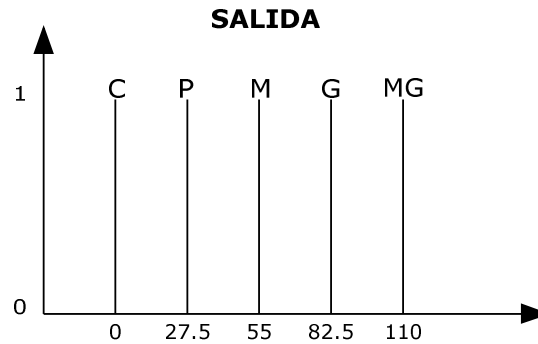
for(i=0;i<5;i++){
    if(i==0&&dvariable<=GraficaDTemp[i][1]||i==4&&dvariable>=GraficaDTemp[i][2
1]){
        PertenenciaDTemp[i]=1; //Valida si dtemp es <A12 o >A52
    }
    else{
        if(dvariable<GraficaDTemp[i][2]&&dvariable>GraficaDTemp[i][0]){ } {
//Valida si dtemp está en el rango (Ax1,Ax3)
        if(dvariable<=GraficaError[i][1]){ } //Valida si dtemp está en el rango
// (Ax1,Ax2)
            PertenenciaDTemp[i]=(dvariable-GraficaDTemp[i][0])/
            (GraficaDTemp[i][1]- GraficaDTemp[i][0]);
        }else{ //Valida si dtemp está en el rango (Ax2,Ax3)
            PertenenciaDTemp[i]=(GraficaDTemp[i][2]-
            dvariable)/(GraficaDTemp[i][2]-GraficaDTemp[i][1]);
        }
    }else{ //Si no está en el rango (Ax1,Ax3) entonces el nivel de
pertenencia
        //es 0
        PertenenciaDTemp[i]=0;
    }
}
}
}
}

```

El método defuzzresistencia() es el encargado de realizar la Defuzzificación, en otras palabras, calcular el valor que el controlador debe entregar según la situación actual.

El universo de discurso de la salida es el siguiente:

Figura 71. Universo de discurso de la salida del controlador.



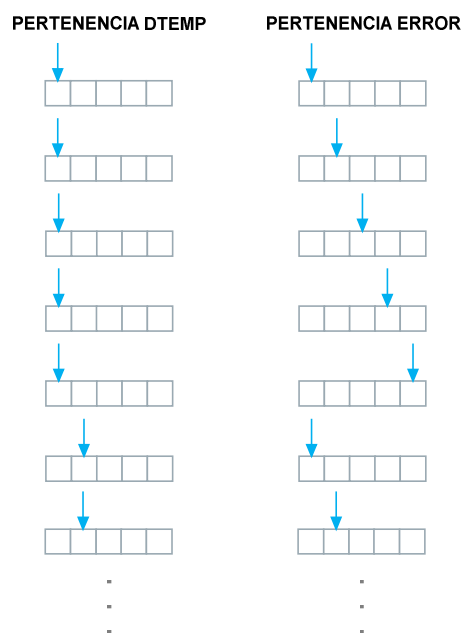
El vector SalidaResistencia de dimensiones 5x5 está definido como variable global y contiene los valores de salida de cada regla. El formato del vector es el siguiente:

Tabla 26. Formato del vector SalidaResistencia.

		Error				
		Ng	Neg	Z	Pos	Pg
DTemp	Ng	C	C	C	MG	MG
	Neg	C	C	C	MG	MG
	Z	C	C	C	M	MG
	Pos	C	C	C	M	G
	Pg	C	C	C	P	M

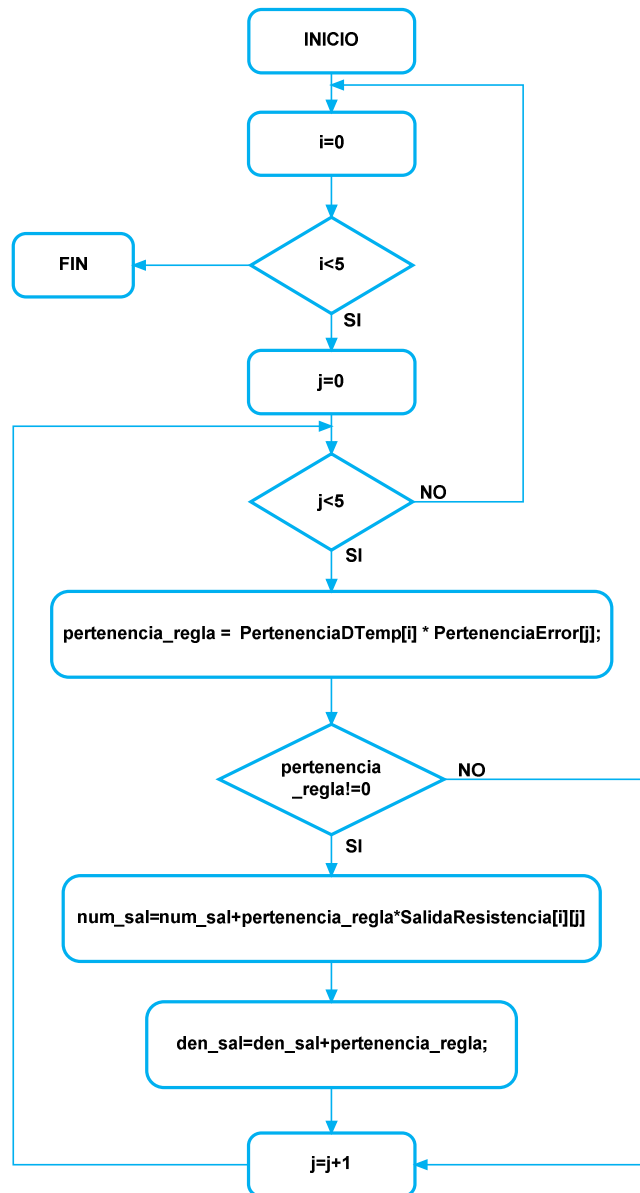
En la Figura 73 se muestra la secuencia que realiza el método defuzz(). Éste método calcula el nivel de pertenencia de cada regla multiplicando cada miembro del vector PertenenciaError con cada miembro del vector PertenenciaDTemp como se muestra en la Figura 72 y almacena el resultado de la operación en la variable pertenencia_regla.

Figura 72. Recorrido de los vectores en el método defuzz().



En caso de que la multiplicación de los niveles de pertenencia de los presentes sea diferente de cero el método incrementará los acumuladores `num_sal` y `den_sal` como se muestra en la Figura 73.

Figura 73. Diagrama de flujo del método defuzz().



El método Angulo_Disparo es el encargado de transformar el valor de tensión entregado por el controlador, en un retardo que comienza a contar cuando se produce un cruce por cero. Los datos de este método se encuentran en la tabla del Anexo 4 en donde muestra el tiempo de carga del Timer 0 para que el porcentaje de la fase que se le entregue a la resistencia sea equivalente al valor calculado por el controlador.

El código del método se muestra a continuación:

```

void Angulo_Disparo(int16 voltaje)
{
    switch(voltaje){
        case 0: val_tmr0=65536;break;
        case 1: val_tmr0=65155;break;
        case 2: val_tmr0=64770;break;
        case 3: val_tmr0=64384;break;
    }
}
  
```

```
case 4: val_tmr0=64014;break;
case 5: val_tmr0=63636;break;
case 6: val_tmr0=63257;break;
case 7: val_tmr0=62882;break;
case 8: val_tmr0=62489;break;
case 9: val_tmr0=62108;break;
case 10: val_tmr0=61742;break;
case 11: val_tmr0=61365;break;
case 12: val_tmr0=60978;break;
case 13: val_tmr0=60611;break;
case 14: val_tmr0=60205;break;
case 15: val_tmr0=59855;break;
case 16: val_tmr0=59466;break;
case 17: val_tmr0=59070;break;
case 18: val_tmr0=58701;break;
case 19: val_tmr0=58327;break;
case 20: val_tmr0=57948;break;
case 21: val_tmr0=57564;break;
case 22: val_tmr0=57176;break;
case 23: val_tmr0=56819;break;
case 24: val_tmr0=56423;break;
case 25: val_tmr0=56060;break;
case 26: val_tmr0=55657;break;
case 27: val_tmr0=55287;break;
case 28: val_tmr0=54915;break;
case 29: val_tmr0=54541;break;
case 30: val_tmr0=54164;break;
case 31: val_tmr0=53785;break;
case 32: val_tmr0=53404;break;
case 33: val_tmr0=53021;break;
case 34: val_tmr0=52636;break;
case 35: val_tmr0=52250;break;
case 36: val_tmr0=51902;break;
case 37: val_tmr0=51513;break;
case 38: val_tmr0=51124;break;
case 39: val_tmr0=50733;break;
case 40: val_tmr0=50381;break;
case 41: val_tmr0=49989;break;
case 42: val_tmr0=49596;break;
case 43: val_tmr0=49243;break;
case 44: val_tmr0=48850;break;
case 45: val_tmr0=48496;break;
case 46: val_tmr0=48103;break;
case 47: val_tmr0=47709;break;
case 48: val_tmr0=47356;break;
case 49: val_tmr0=46963;break;
case 50: val_tmr0=46571;break;
case 51: val_tmr0=46219;break;
case 52: val_tmr0=45828;break;
case 53: val_tmr0=45438;break;
case 54: val_tmr0=45088;break;
case 55: val_tmr0=44701;break;
case 56: val_tmr0=44314;break;
case 57: val_tmr0=43929;break;
case 58: val_tmr0=43545;break;
case 59: val_tmr0=43163;break;
case 60: val_tmr0=42782;break;
case 61: val_tmr0=42404;break;
case 62: val_tmr0=42027;break;
case 63: val_tmr0=41652;break;
case 64: val_tmr0=41280;break;
case 65: val_tmr0=40910;break;
case 66: val_tmr0=40542;break;
case 67: val_tmr0=40140;break;
case 68: val_tmr0=39777;break;
```

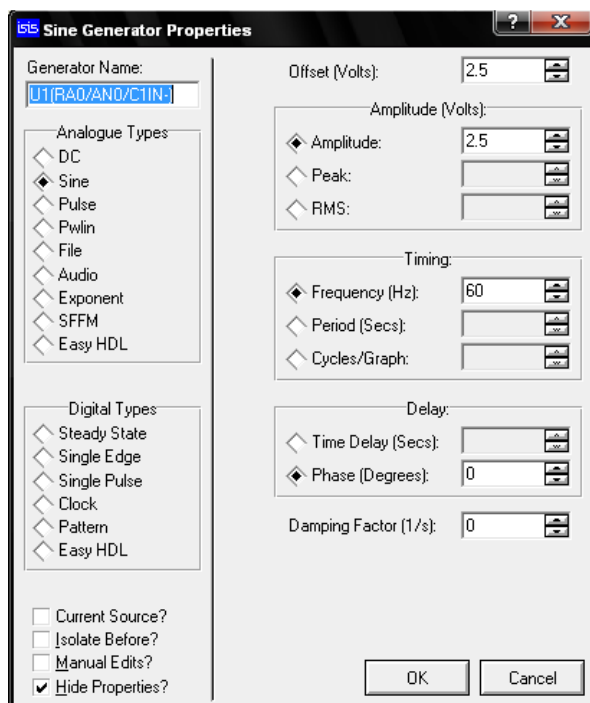
```

case 69: val_tmr0=39382;break;
case 70: val_tmr0=39025;break;
case 71: val_tmr0=38636;break;
case 72: val_tmr0=38250;break;
case 73: val_tmr0=37868;break;
case 74: val_tmr0=37490;break;
case 75: val_tmr0=37117;break;
case 76: val_tmr0=36747;break;
case 77: val_tmr0=36381;break;
case 78: val_tmr0=35987;break;
case 79: val_tmr0=35598;break;
case 80: val_tmr0=35215;break;
case 81: val_tmr0=34837;break;
case 82: val_tmr0=34465;break;
case 83: val_tmr0=34099;break;
case 84: val_tmr0=33708;break;
case 85: val_tmr0=33325;break;
case 86: val_tmr0=32949;break;
case 87: val_tmr0=32580;break;
case 88: val_tmr0=32192;break;
case 89: val_tmr0=31812;break;
case 90: val_tmr0=31440;break;
case 91: val_tmr0=31078;break;
case 92: val_tmr0=30700;break;
case 93: val_tmr0=30309;break;
case 94: val_tmr0=29929;break;
case 95: val_tmr0=29562;break;
case 96: val_tmr0=29185;break;
case 97: val_tmr0=28802;break;
case 98: val_tmr0=28414;break;
case 99: val_tmr0=28044;break;
case 100: val_tmr0=27674;break;
case 101: val_tmr0=27289;break;
case 102: val_tmr0=26912;break;
case 103: val_tmr0=26532;break;
case 104: val_tmr0=26141;break;
case 105: val_tmr0=25775;break;
case 106: val_tmr0=25395;break;
case 107: val_tmr0=25021;break;
case 108: val_tmr0=24642;break;
case 109: val_tmr0=24262;break;
case 110: val_tmr0=23886;break;
}
}

```

Se realizó una simulación en el software ISIS de Proteus 7.6 con el fin de evaluar el rendimiento del método Angulo_Disparo(). El diagrama esquemático de dicha simulación se encuentra en la Figura 74.

Figura 75. Configuración del generador de funciones.

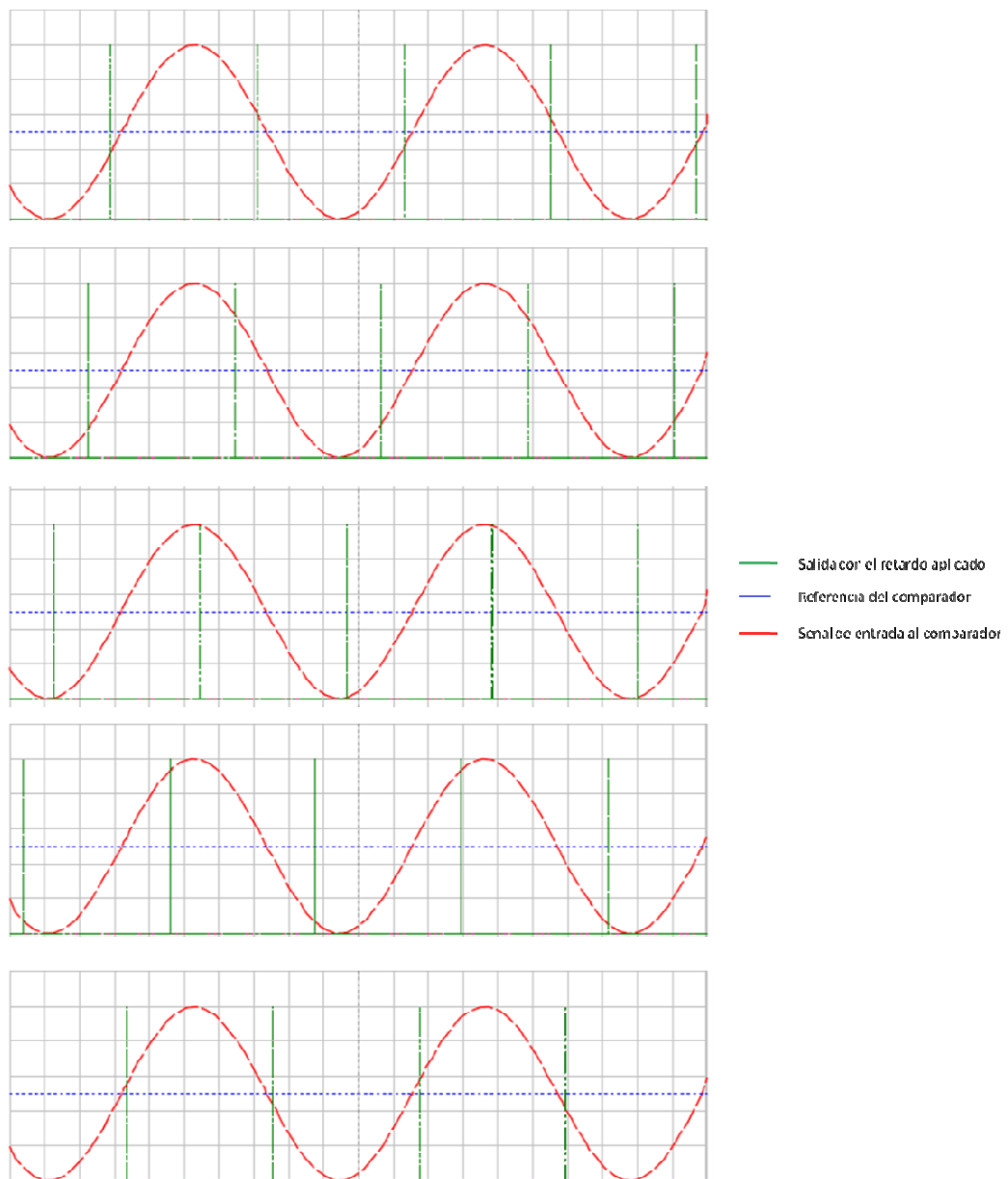


108

se encuentren en el rango $[0,110]$; simultáneamente cada 1ms la tarea `Angulo_Disparo()` monitorea el valor de dicha variable y selecciona de acuerdo a esto el valor de carga que debe tener el Timer 0, guardando dicho valor en la variable `val_tmr0`. Cuando la señal senoidal cruza por la referencia, se carga el valor de la variable `val_tmr0` al Timer 0.

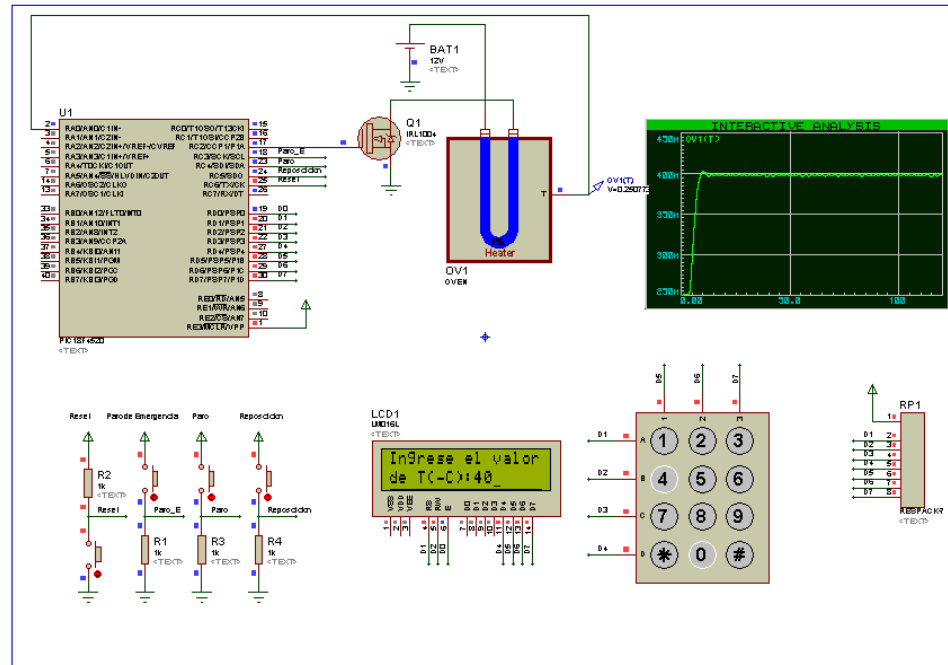
Los resultados de la simulación se muestran en la Figura 76 y en ellos se puede observar que a medida que aumenta el valor de la variable voltaje disminuye el tiempo de retardo para generar el pulso que se utilizará para activar el triac, creando un efecto de barrido donde el pulso se traslada de izquierda a derecha.

Figura 76. Resultados de la simulación de prueba del método `Angulo_Disparo()`.



También se realizó una simulación de la tarea Temp en el software ISIS de Proteus 7.6. El diagrama esquemático de dicha simulación se encuentra en la Figura 77.

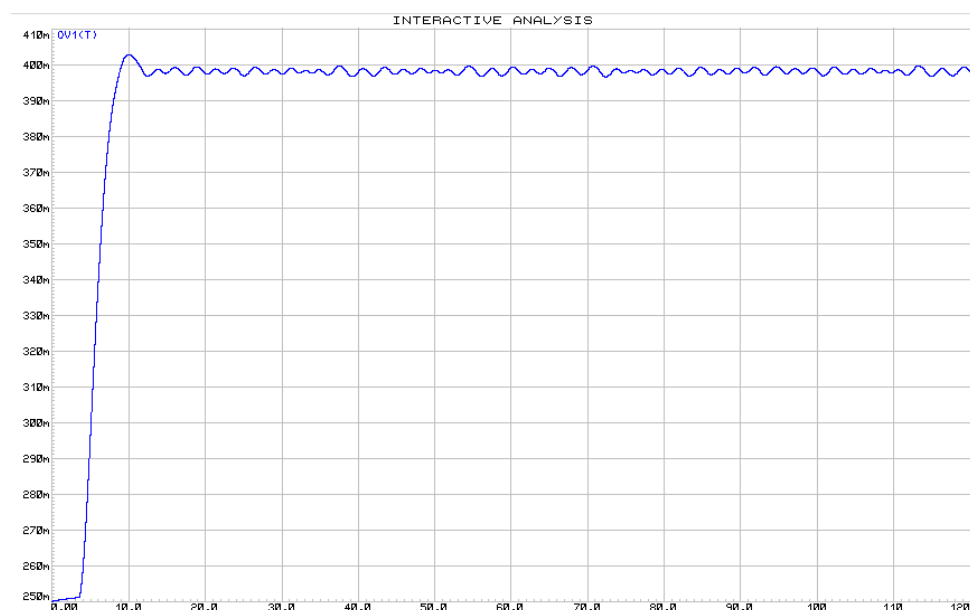
Figura 77. Diagrama esquemático de la simulación de la tarea Temp.



El terminal T que sale del horno y está conectado al pin A0, es un sensor que indica la temperatura del horno y esta programado en 0.001 V/°C.

De la simulación se obtuvo el siguiente comportamiento del horno:

Figura 78. Comportamiento del horno simulado en Proteus 7.6 con controlador difuso.



Podemos apreciar un seguimiento de la referencia con un promedio de error de 0.5 °C, el cuál según las necesidades del cliente es despreciable. El tiempo de estabilización es de 10 segundos lo cuál es bastante complaciente, no obstante se debe recordar que esto depende en gran parte de las propiedades de la resistencia que las propiedades de la resistencia simulada no son las mismas de la resistencia real.

Cabe resaltar que el código con el que se hizo la prueba difiere del código implementado para el control de la resistencia real; entre las diferencias más relevantes se encuentran:

- En la simulación la resistencia está alimentada con DC.
- El control por ángulo de disparo se sustituyó por control PWM.
- Las propiedades de las resistencias son distintas.
- El PIC en la simulación trabaja a 4MHz y la implementación a 40MHz.

El código de la prueba se encuentra en el Anexo 7.

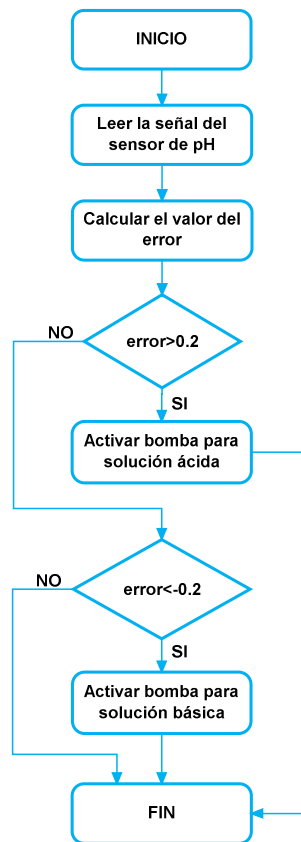
5.3.6.2 Control de pH

Para el control de pH se implementó un control de tipo On-Off ya que debido a la inmensa precisión de las bombas peristálticas, la histéresis puede llegar a ser despreciable.

La tarea pH está encargada de calcular el error de pH en la muestra cada 10ms y dependiendo de si el error es mayor o menor que la referencia decidirá que bomba activar; sin embargo, se añadió una ventana de histéresis de ± 0.2 en el momento de validar el valor del error, esto con el fin de evitar una posible inestabilidad del sistema debido a la inercia que pueda tener la muestra en cuanto al cambio de su nivel de pH.

La secuencia que realiza la tarea pH se encuentra en la Figura 79.

Figura 79. Diagrama de flujo de la tarea pH.



Las variables globales que utiliza la tarea pH son:

```
int secuencia=0;
int secuencia2=0;
float ref=0.0;
float error_pH=-0.0;
```

La rutina de la tarea pH se encuentra a continuación:

```
void pH ( )
{
    set_adc_channel(4); //Selecciona el canal del sensor de pH
    valor = read_adc (); //Lee la señal del sensor
    valor = 1 + valor * 4 / 1024.0; //Se normaliza el la señal medida
    error_pH = ref - valor; //Calcula el error de pH
    if(error_pH>0.2) //Valida si el valor del error de pH es > que la ref
        BombaAcido(); //Si es el caso añade ácido a la muestra
    else if(error_pH<ref-0.2) //Valida si el valor del error de pH es > que la
        ref
        BombaBase(); //Si es el caso base ácido a la muestra
    else
        PORTC=0; //Si no hay error desenergiza las bobinas
    rtos_yield();
}
```

Los métodos bomba base y bomba acido son los encargados de generar las secuencias para el motor correspondiente. Para este caso en particular se implementó una secuencia de medio paso.

La secuencia implementada es la mostrada en la Tabla 27.

Tabla 27. Secuencia de medio paso para motores bipolares.

A	B	C	D
0	1	0	1
0	0	0	1
1	0	0	1
1	0	0	0
1	0	1	0
0	0	1	0
0	1	1	0
0	1	0	0

El código de cada método se encuentra a continuación.

```
void BombaAcido()  
{  switch (secuencia){  
    case 1: portc=0b10100000;  
            secuenciac++;  
            break;  
    case 2: portc=0b10000000;  
            secuenciac++;  
            break;  
    case 3: portc=0b10010000;  
            secuenciac++;  
            break;  
    case 4: portc=0b00010000;  
            secuenciac++;  
            break;  
    case 5: portc=0b01010000;  
            secuenciac++;  
            break;  
    case 6: portc=0b01000000;  
            secuenciac++;  
            break;  
    case 7: portc=0b01100000;  
            secuenciac++;  
            break;  
    case 8: portc=0b00100000;  
            secuenciac=1;  
            break;  
  }  
}  
  
void BombaBase()  
{  switch (secuencia2){  
    case 1: portc=0b00001010;  
            secuenciab++;  
            break;  
    case 2: portc=0b00001000;  
            secuenciab++;  
            break;  
    case 3: portc=0b00001001;  
            secuenciab++;  
            break;  
    case 4: portc=0b00000001;
```

```

        secuencia2++;
        break;
case 5: portc=0b00000101;
        secuencia2++;
        break;
case 6: portc=0b00000100;
        secuencia2++;
        break;
case 7: portc=0b00000110;
        secuencia2++;
        break;
case 8: portc=0b00000010;
        secuencia2=1;
        break;
    }
}

```

Como se puede apreciar estos métodos simplemente se encargan de sacar el valor correspondiente a la secuencia que necesita cada motor por el puerto C.

6 CONCLUSIONES

- Se diseñó la automatización de los biorreactores de laboratorio, equipos experimentales para la producción de etanol a partir de yuca y batata.
- Se analizaron las variables a controlar, y se determinó que sí son dependientes.
- Se diseñaron los controladores de pH y temperatura para los biorreactores.
- Se realizó un modelado CAD del dispositivo automatizado.
- Se realizó un presupuesto para el sistema de control.
- Se Simuló el comportamiento de los actuadores para probar los controladores.

7 BIBLIOGRAFÍA

1. ABB. *Sensores combinados de pH/Redox (ORP) Serie AP300*. [En línea] [Citado el: 10 de 01 de 2010.] [http://www05.abb.com/global/scot/scot203.nsf/veritydisplay/cf33e1ec7e8aee00c12575c8003f4a49/\\$File/SS_AP300-E_4.pdf](http://www05.abb.com/global/scot/scot203.nsf/veritydisplay/cf33e1ec7e8aee00c12575c8003f4a49/$File/SS_AP300-E_4.pdf).
2. Blogspot. [En línea] [Citado el: 05 de 01 de 2010.] <http://losmicrocontroladores.blogspot.com/2008/12/microcontroladores-historias.html>.
3. Blogspot. [En línea] [Citado el: 11 de 02 de 2010.] http://4.bp.blogspot.com/_js6wgtUcfdQ/R4O4QP0aVRI/AAAAAAAAABqY/-Ei4TxsbBKc/s400/memoria_eprom_27C256_32K.jpg.
4. **Castillo, Omar. Villegas, Baryeric.** Blogspot. *Sensores potenciométricos*. [En línea] [Citado el: 10 de 02 de 2010.] <http://sensorpotenciometrico.blogspot.com/>.
5. Design. *Electrodos industriales pH - ORP*. [En línea] [Citado el: 03 de 01 de 2010.] <http://www.desin.com/principal.asp>.
6. Design. *Transmisor DMM4000-pH*. [En línea] [Citado el: 10 de 01 de 2010.] <http://www.desin.com/principal.asp>.
7. Foros de Electronica. [En línea] [Citado el: 03 de 01 de 2010.] <http://www.forosdeelectronica.com/f19/introduccion-motores-paso-paso-289/>.
8. **Pacinno, Kevin M.** *Fuzzy Control*. Menlo Park, California : Addison-Wesley, 1988. págs. 86-87. ISBN 0-201-18074-X.
9. GUASCH. *Componentes y electronica de potencia*. [En línea] [Citado el: 02 de 01 de 2010.] http://www.e-guasch.com/onlinedocs/catalogue/appnotes/AN080917_CONTROL%20VS%20ANGULO%20DE%20FASE%20EN%20REGULADORES%20AC.pdf.
10. HVW Technologies. *PIC 18F4520-I/P*. [En línea] [Citado el: 10 de 01 de 2010.] http://www.hvwtech.com/products_view.asp?ProductID=883.
11. HVW Technologies. *PIC 16F877A/P*. [En línea] [Citado el: 10 de 01 de 2010.] http://www.hvwtech.com/products_view.asp?ProductID=883.
12. **Pérez Villa, José Fernando.** *Introducción a los Microprocesadores y Microcontroladores*. [Documento PDF] Santiago de Cali, Colombia : s.n., 2007.
13. La Salle. *Universidad Ramón Llull*. [En línea] [Citado el: 11 de 01 de 2010.] <http://www.salle.url.edu/portal/lasalle/enginyeria-arquitectura/Controller?mvchandler=portals&action=change-language&view=collage&lang=es>.
14. **Muquillaza, Rubén.** Monografias.com. [En línea] [Citado el: 17 de 01 de 2010.] <http://www.monografias.com/trabajos37/motores/motores2.shtml>.

15. Omega. [En línea] [Citado el: 10 de 01 de 2010.] <http://www.omega.com/Green/pdf/PHTX271.pdf>.
16. Qmax. [En línea] [Citado el: 15 de 01 de 2010.] <http://www.qmax.com.ar/NotasAplicacion/NA4.pdf>.
17. **Carletti, Eduardo**. Robots Argentina. [En línea] [Citado el: 15 de 01 de 2010.] http://robots-argentina.com.ar/MotorPP_basico.htm.
18. Scribd. *Control Difuso*. [En línea] [Citado el: 01 de 02 de 2010.] <http://www.scribd.com/doc/3200501/Control-Difuso>.
19. Thomas Industries. [En línea] [Citado el: 10 de 01 de 2010.] <http://www.gd-thomas.com/workareashowcontent.aspx?id=22882>.
20. **López, Jesús Alfonso**. Tripod. [En línea] [Citado el: 16 de 01 de 2010.] http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html.
21. **Martínez, Mario**. Universidad Autónoma de Nuevo León. [En línea] [Citado el: 17 de 01 de 2010.] <http://www.mecanica-uanl.com/pdfs/Stepper%20Motors.pdf>.
22. Verder Flex. [En línea] [Citado el: 10 de 01 de 2010.] <http://www.verderflex.com/file.php/281/M045%5B1%5D.pdf>.
23. Wikipedia.es. *La enciclopedia libre*. [En línea] [Citado el: 17 de 01 de 2010.] http://es.wikipedia.org/wiki/Motor_paso_a_paso.
24. Wikipedia.es. *La enciclopedia libre*. [En línea] [Citado el: 03 de 02 de 2010.] http://es.wikipedia.org/wiki/Motor_de_reluctancia_variable.
25. Wikipedia.es. *La enciclopedia libre*. [En línea] [Citado el: 06 de 01 de 2010.] <http://es.wikipedia.org/wiki/Biorreactor>.
26. Wikipedia.es. *La enciclopedia libre*. [En línea] [Citado el: 15 de 02 de 2010.] http://es.wikipedia.org/wiki/Conversor_anal%C3%B3gico-digital.
27. Wikipedia.es. *La enciclopedia libre*. [En línea] [Citado el: 02 de 01 de 2010.] <http://es.wikipedia.org/wiki/Microcontrolador>.

8 ANEXOS

Anexo 1. ESQUEMA DEL CIRCUITO ELECTRÓNICO

Anexo 2. CÓDIGO PRUEBA ÁNGULO DISPARO

```
#include <18F4520.h>
#device adc=10

#FUSES NOWDT
#FUSES XT
#FUSES PUT
#use delay(clock=4M)
#use rtos(timer=1)
#use standard_io(c)

#byte CMCON=0xFB4
#byte TRISA=0XF92
#byte TRISC=0XF94
#byte TRISD=0XF95
#byte PORTC=0XF82
#byte PORTD=0XF83

#task(rate=1ms,max=1ms)
void Angulo_Disparo();

#task(rate=5ms,max=1ms)
void Valor();

/*****VARIABLES GLOBALES*****/
int val_tmr0=0;
int voltaje=0;
/*****INTERRUPCIONES*****/
#int_TIMER0
void TIMER0_isr()
{
    output_high(PIN_C3);
    delay_us(10);
    output_low(PIN_C3);
}

#int_COMP
void COMP_isr()
{
    if(C1OUT){
        set_timer0(val_tmr0);
    }else if(!C1OUT){
        set_timer0(val_tmr0);
    }
}

void Valor()
{
    if(voltaje<=110)
        voltaje++;
    else{
        voltaje=0;
    }
    rtos_yield();
}

void Angulo_Disparo()
{
    switch(voltaje){
        case 0 :val_tmr0= 0 ;break;
        case 1 :val_tmr0= 0 ;break;
        case 2 :val_tmr0= 0 ;break;
        case 3 :val_tmr0= 3 ;break;
    }
```



```

case 4 :val_tmr0= 5 ;break;
case 5 :val_tmr0= 8 ;break;
case 6 :val_tmr0= 10 ;break;
case 7 :val_tmr0= 12 ;break;
case 8 :val_tmr0= 15 ;break;
case 9 :val_tmr0= 17 ;break;
case 10 :val_tmr0= 19 ;break;
case 11 :val_tmr0= 22 ;break;
case 12 :val_tmr0= 24 ;break;
case 13 :val_tmr0= 26 ;break;
case 14 :val_tmr0= 29 ;break;
case 15 :val_tmr0= 31 ;break;
case 16 :val_tmr0= 34 ;break;
case 17 :val_tmr0= 36 ;break;
case 18 :val_tmr0= 38 ;break;
case 19 :val_tmr0= 41 ;break;
case 20 :val_tmr0= 43 ;break;
case 21 :val_tmr0= 46 ;break;
case 22 :val_tmr0= 48 ;break;
case 23 :val_tmr0= 50 ;break;
case 24 :val_tmr0= 53 ;break;
case 25 :val_tmr0= 55 ;break;
case 26 :val_tmr0= 57 ;break;
case 27 :val_tmr0= 60 ;break;
case 28 :val_tmr0= 62 ;break;
case 29 :val_tmr0= 64 ;break;
case 30 :val_tmr0= 67 ;break;
case 31 :val_tmr0= 69 ;break;
case 32 :val_tmr0= 72 ;break;
case 33 :val_tmr0= 74 ;break;
case 34 :val_tmr0= 76 ;break;
case 35 :val_tmr0= 79 ;break;
case 36 :val_tmr0= 81 ;break;
case 37 :val_tmr0= 83 ;break;
case 38 :val_tmr0= 86 ;break;
case 39 :val_tmr0= 88 ;break;
case 40 :val_tmr0= 90 ;break;
case 41 :val_tmr0= 93 ;break;
case 42 :val_tmr0= 95 ;break;
case 43 :val_tmr0= 98 ;break;
case 44 :val_tmr0= 100 ;break;
case 45 :val_tmr0= 102 ;break;
case 46 :val_tmr0= 105 ;break;
case 47 :val_tmr0= 107 ;break;
case 48 :val_tmr0= 109 ;break;
case 49 :val_tmr0= 112 ;break;
case 50 :val_tmr0= 114 ;break;
case 51 :val_tmr0= 116 ;break;
case 52 :val_tmr0= 119 ;break;
case 53 :val_tmr0= 121 ;break;
case 54 :val_tmr0= 123 ;break;
case 55 :val_tmr0= 126 ;break;
case 56 :val_tmr0= 128 ;break;
case 57 :val_tmr0= 131 ;break;
case 58 :val_tmr0= 133 ;break;
case 59 :val_tmr0= 136 ;break;
case 60 :val_tmr0= 138 ;break;
case 61 :val_tmr0= 140 ;break;
case 62 :val_tmr0= 143 ;break;
case 63 :val_tmr0= 145 ;break;
case 64 :val_tmr0= 147 ;break;
case 65 :val_tmr0= 150 ;break;
case 66 :val_tmr0= 152 ;break;
case 67 :val_tmr0= 154 ;break;
case 68 :val_tmr0= 157 ;break;

```

```

        case 69 :val_tmr0= 159 ;break;
        case 70 :val_tmr0= 161 ;break;
        case 71 :val_tmr0= 164 ;break;
        case 72 :val_tmr0= 166 ;break;
        case 73 :val_tmr0= 169 ;break;
        case 74 :val_tmr0= 171 ;break;
        case 75 :val_tmr0= 173 ;break;
        case 76 :val_tmr0= 176 ;break;
        case 77 :val_tmr0= 178 ;break;
        case 78 :val_tmr0= 180 ;break;
        case 79 :val_tmr0= 183 ;break;
        case 80 :val_tmr0= 185 ;break;
        case 81 :val_tmr0= 188 ;break;
        case 82 :val_tmr0= 190 ;break;
        case 83 :val_tmr0= 192 ;break;
        case 84 :val_tmr0= 195 ;break;
        case 85 :val_tmr0= 197 ;break;
        case 86 :val_tmr0= 199 ;break;
        case 87 :val_tmr0= 202 ;break;
        case 88 :val_tmr0= 204 ;break;
        case 89 :val_tmr0= 206 ;break;
        case 90 :val_tmr0= 209 ;break;
        case 91 :val_tmr0= 211 ;break;
        case 92 :val_tmr0= 213 ;break;
        case 93 :val_tmr0= 216 ;break;
        case 94 :val_tmr0= 218 ;break;
        case 95 :val_tmr0= 221 ;break;
        case 96 :val_tmr0= 223 ;break;
        case 97 :val_tmr0= 225 ;break;
        case 98 :val_tmr0= 228 ;break;
        case 99 :val_tmr0= 230 ;break;
        case 100 :val_tmr0= 232 ;break;
        case 101 :val_tmr0= 235 ;break;
        case 102 :val_tmr0= 237 ;break;
        case 103 :val_tmr0= 239 ;break;
        case 104 :val_tmr0= 242 ;break;
        case 105 :val_tmr0= 244 ;break;
        case 106 :val_tmr0= 247 ;break;
        case 107 :val_tmr0= 249 ;break;
        case 108 :val_tmr0= 251 ;break;
        case 109 :val_tmr0= 254 ;break;
        case 110 :val_tmr0= 255 ;break;
    }
    rtos_yield();
}

/*****MAIN*****/
*/
void main()
{
    TRISA=0b00001001;
    TRISC=0;
    PORTC=0;
    TRISD=0;
    CMCON = 0b00000001;
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_32|RTCC_8_bit);
    setup_adc_ports(AN0_TO_AN4|VREF_VDD);
    setup_adc(ADC_CLOCK_INTERNAL|ADC_TAD_MUL_6);
    enable_interrupts( INT_COMP );
    enable_interrupts( INT_TIMER0 );
    enable_interrupts( GLOBAL );
    rtos_run();
}

```

Anexo 3. CÓDIGO PARA LA PRUEBA DE LA TAREA TEMP

```
#INCLUDE <18f4520.h>
#device adc=10

#FUSES NOWDT                      //No Watch Dog Timer
#FUSES XT                        //Resistor/Capacitor Osc with CLKOUT
#FUSES PUT                      //Power Up Timer
#FUSES NOPROTECT                //Code not protected from reading

#use delay(clock=400000)
#use rtos(timer=0)              //Directiva del RTOS

#include "D:\Trabajo de Grado\Codigo PICC\Fuzzy Menu\Control Fuzzy.h"
#include "D:\Trabajo de Grado\Codigo PICC\Fuzzy Menu\LCD.C"
#include "D:\Trabajo de Grado\Codigo PICC\Fuzzy Menu\KBD.C"
#Byte TRISC = 0x0f94
#Byte TRISD = 0x0f95
#Byte PORTC = 0x0f82

float valor,x;
int16 control;
int ref;
float dtemp=0.0;
float error=0.0;
float valor_1=0.0;
float kd,k,ko;
float PertenenciaError[5];
float PertenenciaDTemp[5];

const float SalidaResistencia[5][5]={0 , 0 , 0 , 900 , 900},
                                       {0 , 0 , 0 , 900 , 900},
                                       {0 , 0 , 0 , 900 , 900},
                                       {0 , 0 , 0 , 450 , 900},
                                       {0 , 0 , 0 , 225 , 900}};

#task(rate=12ms, max=10ms)
Void fuzzy( );

#task(rate=1ms, max=1ms)
Void Paro_Emergencia( );

#task(rate=1ms, max=1ms)
Void Reset( );

#task(rate=1ms, max=1ms)
Void Paro( );

#task(rate=1ms, max=1ms)
Void Reposicion( );

#task(rate=1ms, max=1ms)
Void Display( );

/*****METODOS*****/
*/

void lcd_cursor_on()
{
    lcd.rs=0;
    lcd_send_nibble(0);
    lcd_send_nibble(0x0F);
}
```

```

char kbd_get_key()
{
    char key='\0';
    while(key=='\0')
        key=kbd_getc();
    return key;
}

void Bienvenida()
{
    printf(lcd_putc,"Bienvenido");
    delay_ms(500);
    lcd_putc('.');
    delay_ms(500);
    lcd_putc('.');
    delay_ms(500);
    lcd_putc('.');
    delay_ms(500);
    lcd_cursor_on();
}

int get_ref_Temp()
{
    int datos[3];
    int i,key;
    int ref_temp;
    printf(lcd_putc,"\fIngrese el valor\nde T(°C):__\b\b");
    for(i=0;i<=2;){
        key=kbd_get_key();
        datos[2]=key-0x30;
        if(key!='*'&&key!='#'){
            if(i==0||i==1){
                lcd_putc(key);
                datos[i]=key-0x30;
                i++;
            }
        }else if(key=='*'){
            switch(i){
                case 0: break;
                case 1: printf(lcd_putc, "\b\b");
                        i--;
                        break;
                case 2: printf(lcd_putc, "\b\b");
                        i--;
                        break;
            }
        }else if(key=='#'){
            switch(i){
                case 0: break;
                case 1: printf(lcd_putc, "0");
                        datos[1]=0;
                        i=2;
                        break;
                case 2: lcd_putc('\f');
                        i=3;
                        break;
            }
        }
    }
    return ref_Temp=datos[0]*10+datos[1];
}

void CalcularPertenencia(float error,float dvariable,)
{
    int i;

```

```

const float GraficaDTemp[5][3]={ { -1 , -1 , -0.5},
                                   { -1 , -0.5 , 0 },
                                   { -0.5 , 0 , 0.5},
                                   { 0 , 0.5 , 1 },
                                   { 0.5 , 1 , 1 }};

const float GraficaError[5][3]={ {-100 , -100 , -25},
                                   { -50 , -25 , 0 },
                                   { -25 , 0 , 25},
                                   { 0 , 20 , 50},
                                   { 25 , 100 , 100}};

for(i=0;i<5;i++){
    if (i==0 && error<=GraficaError[i][1]||i==4 &&
error>=GraficaError[i][2]){
        PertenenciaError[i]=1;
    }
    else{
        if(error<GraficaError[i][2]&&error>GraficaError[i][0]){
            if(error<=GraficaError[i][1]){
                PertenenciaError[i]=(error-
GraficaError[i][0])/(GraficaError[i][1]-GraficaError[i][0]);
            }else{
                PertenenciaError[i]=(GraficaError[i][2]-
error)/(GraficaError[i][2]-GraficaError[i][1]);
            }
        }else{
            PertenenciaError[i]=0;
        }
    }
}

for(i=0;i<5;i++){
    if (i==0 && dvariable<=GraficaDTemp[i][1]||i==4 &&
dvariable>=GraficaDTemp[i][2]){
        PertenenciaDTemp[i]=1;
    }
    else{
        if(dvariable<GraficaDTemp[i][2]&&dvariable>GraficaDTemp[i][0]){
            if(dvariable<=GraficaError[i][1]){
                PertenenciaDTemp[i]=(dvariable-
GraficaDTemp[i][0])/(GraficaDTemp[i][1]-GraficaDTemp[i][0]);
            }else{
                PertenenciaDTemp[i]=(GraficaDTemp[i][2]-
dvariable)/(GraficaDTemp[i][2]-GraficaDTemp[i][1]);
            }
        }else{
            PertenenciaDTemp[i]=0;
        }
    }
}

float defuzzResistencia()
{
    int i,j;
    float num_sal=0;
    float den_sal=0;
    float pertenencia_regla = 0;

    for(i=0;i<5;i++){
        for(j=0;j<5;j++){
            pertenencia_regla = PertenenciaDTemp[i] * PertenenciaError[j];
            if(pertenencia_regla!=0){

```

```

        num_sal=num_sal+pertenencia_regla*SalidaResistencia[i][j];
        den_sal=den_sal+pertenencia_regla;
    }
}
}return num_sal / den_sal;
}

/*****TAREAS*****/

void Display()
{
    lcd_gotoxy(1,2);
    printf(lcd_putc, "Temp = %2.1F",valor);
    rtos_yield();
}

void Paro_Emergencia()
{
    if(input_state(PIN_C3)==1){
        rtos_disable(fuzzy);
        set_pwm1_duty(0);
        rtos_disable(Paro);
        rtos_disable(Reposcicion);
    }
    rtos_yield();
}

void Reset()
{
    if(input_state(PIN_C6)==0){
        rtos_terminate();
    }
    rtos_yield();
}

void Paro()
{
    if(input_state(PIN_C4)==1){
        rtos_disable(fuzzy);
        set_pwm1_duty(0);
    }
    rtos_yield();
}

void Reposcicion()
{
    if(input_state(PIN_C5)==1){
        rtos_enable(fuzzy);
    }
    rtos_yield();
}

void fuzzy ( )
{
    k = 0.5;
    kd = 5;
    ko = 2.7;
    set_adc_channel(0);
    valor = read_adc ();
    valor *= 500.0 / 1024.0;
    error = k * (ref - valor);
    dtemp = kd * (valor - valor_1);
    CalcularPertenencia(error,dtemp);
    x = defuzzResistencia();
    x*=ko;
    control = x;
}

```

```

        if (control>900) control=900;
//    if (control<0)    control=0;
    set_pwm1_duty(control);
    valor_1=valor;
    rtos_yield();
}

void main()
{
    TRISC=0b011111000;
    TRISD=0;
    PORTC=0;
    setup_timer_2(t2_div_by_4,249,1);
    setup_ccp1(ccp_pwm);
    set_pwm1_duty(0);
    setup_adc_ports(all_analog);
    setup_adc(ADC_CLOCK_INTERNAL);
    lcd_init();
    Bienvenida();
    kbd_init();
    while(true){
        lcd_gotoxy(1,1);
        ref=get_ref_Temp();
        printf(lcd_putc,"Ref Temp = %2u",ref);
        rtos_run();
    }
}

```

Anexo 4. TABLA DE DATOS PARA LA TAREA ÁNGULO DISPARO

Angulo	Valor	Voltaje RMS	Tiempo	Carga TMRO	Cuenta Real
180	0	0	0,008330	23886	0,008330
175,8	0,009138	1	0,008254	24267	0,008254
173,3	0,018396	2	0,008177	24652	0,008177
171,2	0,027663	3	0,008100	25038	0,008100
169,4	0,036532	4	0,008026	25408	0,008026
167,7	0,045609	5	0,007950	25786	0,007950
166,1	0,054722	6	0,007874	26165	0,007874
164,6	0,063729	7	0,007799	26540	0,007799
163,1	0,073155	8	0,007721	26933	0,007721
161,7	0,082308	9	0,007644	27314	0,007644
160,4	0,091095	10	0,007571	27680	0,007571
159,1	0,100146	11	0,007496	28057	0,007496
157,8	0,109446	12	0,007418	28444	0,007418
156,6	0,118241	13	0,007345	28811	0,007345
155,3	0,127986	14	0,007264	29217	0,007264
154,2	0,136399	15	0,007194	29567	0,007194
153	0,145743	16	0,007116	29956	0,007116
151,8	0,155252	17	0,007037	30352	0,007037
150,7	0,164107	18	0,006963	30721	0,006963
149,6	0,173087	19	0,006888	31095	0,006888
148,5	0,182186	20	0,006812	31474	0,006812
147,4	0,1914	21	0,006736	31858	0,006736
146,3	0,200721	22	0,006658	32246	0,006658
145,3	0,209283	23	0,006587	32603	0,006587
144,2	0,218793	24	0,006507	32999	0,006507
143,2	0,227519	25	0,006435	33362	0,006435
142,1	0,237199	26	0,006354	33765	0,006354
141,1	0,24607	27	0,006280	34135	0,006280
140,1	0,255004	28	0,006206	34507	0,006206
139,1	0,263997	29	0,006131	34881	0,006131
138,1	0,273045	30	0,006056	35258	0,006056
137,1	0,282145	31	0,005980	35637	0,005980
136,1	0,291292	32	0,005904	36018	0,005904
135,1	0,300484	33	0,005827	36401	0,005827
134,1	0,309715	34	0,005750	36786	0,005750
133,1	0,318984	35	0,005673	37172	0,005673
132,2	0,327353	36	0,005603	37520	0,005603
131,2	0,336681	37	0,005525	37909	0,005525
130,2	0,346035	38	0,005448	38298	0,005448

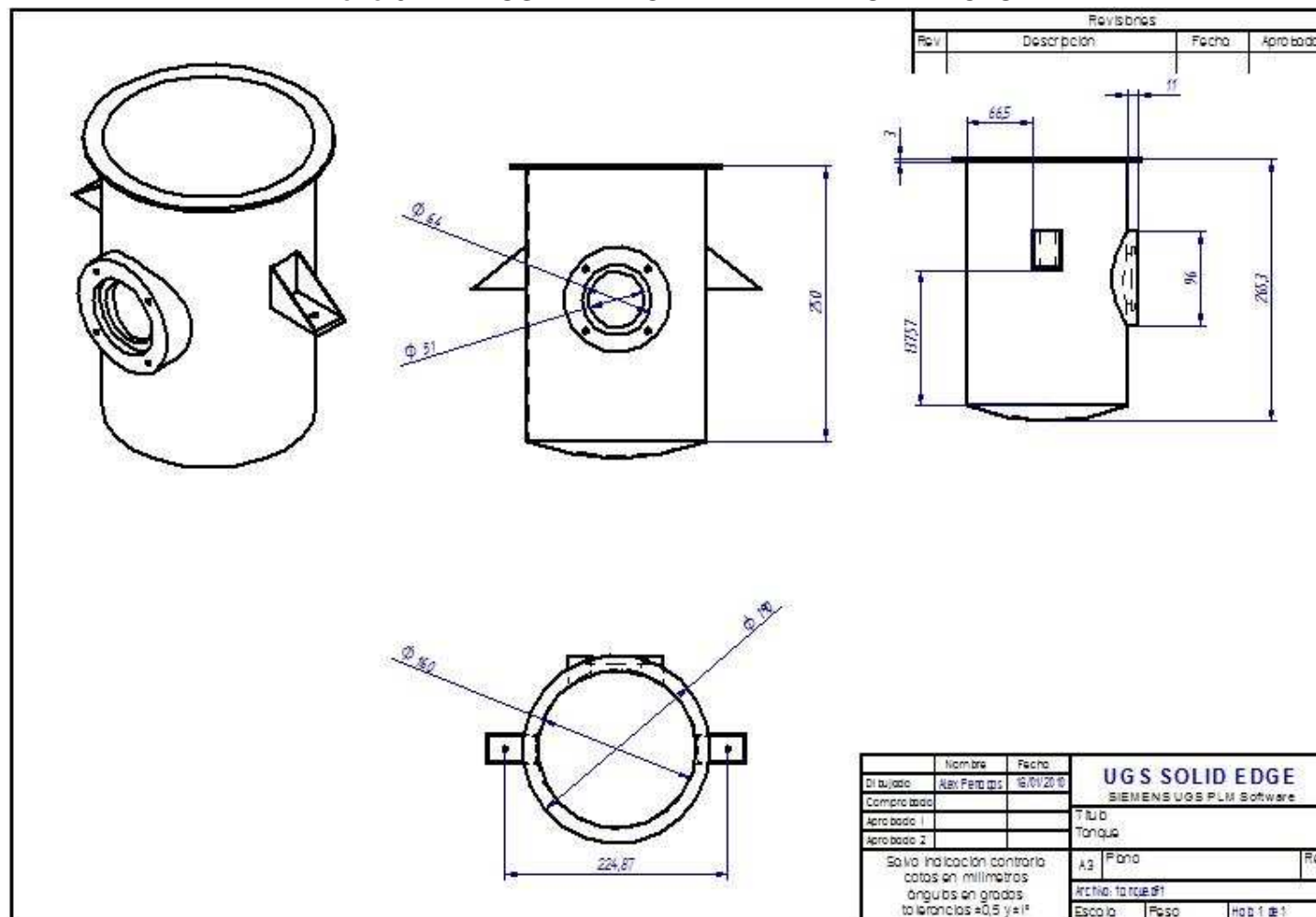
Anexo 4. (Continuación).

Angulo	Valor	Voltaje RMS	Tiempo	Carga TMRO	Cuenta Real
129,2	0,355412	39	0,005369	38689	0,005369
128,3	0,363868	40	0,005299	39041	0,005299
127,3	0,373279	41	0,005221	39433	0,005221
126,3	0,382703	42	0,005142	39826	0,005142
125,4	0,391192	43	0,005071	40179	0,005071
124,4	0,400631	44	0,004993	40572	0,004993
123,5	0,409128	45	0,004922	40926	0,004922
122,5	0,41857	46	0,004843	41319	0,004843
121,5	0,428009	47	0,004765	41713	0,004765
120,6	0,436499	48	0,004694	42066	0,004694
119,6	0,445924	49	0,004615	42459	0,004615
118,6	0,455336	50	0,004537	42851	0,004537
117,7	0,463794	51	0,004467	43203	0,004467
116,7	0,473175	52	0,004388	43594	0,004388
115,7	0,482534	53	0,004310	43984	0,004310
114,8	0,490938	54	0,004240	44334	0,004240
113,8	0,500249	55	0,004163	44721	0,004163
112,8	0,509531	56	0,004086	45108	0,004086
111,8	0,51878	57	0,004009	45493	0,004009
110,8	0,527994	58	0,003932	45877	0,003932
109,8	0,537169	59	0,003855	46259	0,003855
108,8	0,546304	60	0,003779	46640	0,003779
107,8	0,555395	61	0,003704	47018	0,003704
106,8	0,564439	62	0,003628	47395	0,003628
105,8	0,573435	63	0,003553	47770	0,003553
104,8	0,582378	64	0,003479	48142	0,003479
103,8	0,591268	65	0,003405	48512	0,003405
102,8	0,6001	66	0,003331	48880	0,003331
101,7	0,609747	67	0,003251	49282	0,003251
100,7	0,618453	68	0,003178	49645	0,003178
99,6	0,627954	69	0,003099	50040	0,003099
98,6	0,636521	70	0,003028	50397	0,003028
97,5	0,645865	71	0,002950	50786	0,002950
96,4	0,655121	72	0,002873	51172	0,002873
95,3	0,664288	73	0,002796	51554	0,002796
94,2	0,673362	74	0,002721	51932	0,002721
93,1	0,682341	75	0,002646	52305	0,002646
92	0,691221	76	0,002572	52675	0,002572
90,9	0,700001	77	0,002499	53041	0,002499

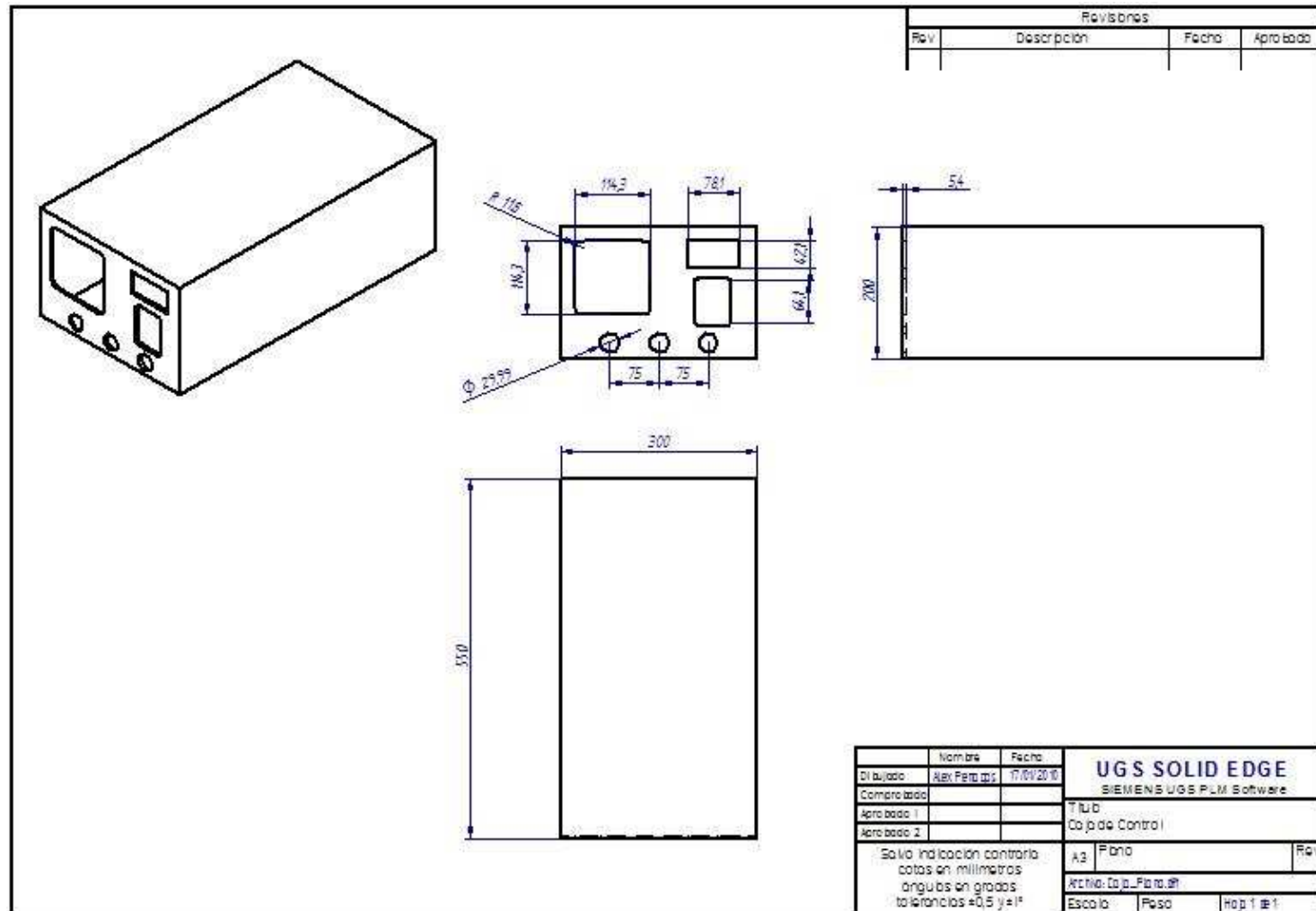
Anexo 4. (Continuación).

Angulo	Valor	Voltaje RMS	Tiempo	Carga TMR0	Cuenta Real
89,7	0,70946	78	0,002420	53435	0,002420
88,5	0,718793	79	0,002342	53824	0,002342
87,3	0,727996	80	0,002266	54207	0,002266
86,1	0,737066	81	0,002190	54585	0,002190
84,9	0,746001	82	0,002116	54957	0,002116
83,7	0,754797	83	0,002043	55323	0,002043
82,4	0,764167	84	0,001964	55714	0,001964
81,1	0,773368	85	0,001888	56097	0,001888
79,8	0,782396	86	0,001813	56473	0,001813
78,5	0,791249	87	0,001739	56842	0,001739
77,1	0,800585	88	0,001661	57230	0,001661
75,7	0,80971	89	0,001585	57610	0,001585
74,3	0,818623	90	0,001511	57982	0,001511
72,9	0,82732	91	0,001438	58344	0,001438
71,4	0,836396	92	0,001363	58722	0,001363
69,8	0,845798	93	0,001285	59113	0,001285
68,2	0,854908	94	0,001209	59493	0,001209
66,6	0,863724	95	0,001135	59860	0,001135
64,9	0,872766	96	0,001060	60237	0,001060
63,1	0,881973	97	0,000983	60620	0,000983
61,2	0,891279	98	0,000906	61008	0,000906
59,3	0,900162	99	0,000832	61378	0,000832
57,3	0,909052	100	0,000758	61748	0,000758
55,1	0,918289	101	0,000681	62133	0,000681
52,8	0,92734	102	0,000605	62510	0,000605
50,3	0,93648	103	0,000529	62890	0,000529
47,5	0,945865	104	0,000451	63281	0,000451
44,6	0,954653	105	0,000378	63647	0,000378
41,2	0,963778	106	0,000302	64027	0,000302
37,3	0,972738	107	0,000227	64401	0,000227
32,4	0,981839	108	0,000151	64780	0,000151
25,5	0,990969	109	0,000075	65160	0,000075
0	1	110	0,000000	65536	0,000000

Anexo 5. PLANOS DEL RECIPIENTE DEL BIORREACTOR



Anexo 6. PLANOS DE LA CARCAZA DEL DISPOSITIVO DE CONTROL



Anexo 7. CÓDIGO IMPLEMENTADO EN EL SISTEMA DE CONTROL

```
#INCLUDE <18f4520.h>
#device adc=10

#FUSES NOWDT                      //No Watch Dog Timer
#FUSES XT                          //Resistor/Capacitor Osc with CLKOUT
#FUSES PUT                          //Power Up Timer
#FUSES NOPROTECT                  //Code not protected from reading

#use delay(clock=400000)
#use rtos(timer=0)                //Directiva del RTOS

#include "D:\Trabajo de Grado\Codigo PICC\Trabajo de Grado\Trabajo de
Grado.h"
#include "D:\Trabajo de Grado\Codigo PICC\Trabajo de Grado\LCD.C"
#include "D:\Trabajo de Grado\Codigo PICC\Trabajo de Grado\KBD.C"
#Byte TRISA = 0x0f92
#Byte TRISB = 0x0f93
#Byte TRISC = 0x0f94
#Byte TRISD = 0x0f95
#Byte TRISE = 0x0f96
#Byte PORTA = 0x0f80
#Byte PORTB = 0x0f81
#Byte PORTC = 0x0f82
#Byte PORTD = 0x0f83
#Byte PORTE = 0x0f84
#Byte CMCON = 0xFB4

//CONSTANTES DEL CONTROLADOR DE TEMPERATURA
int ref;
int16 control;
float valor,x;
float dtemp = 0.0;
float error;
float valor_1 = 0.0;
float k = 1;
float kd = 5;
float ko = 1.35;
float PertenenciaError_temp[5];
float PertenenciaDTemp[5];

//CONSTANTES DEL CONTROLADOR DE pH
int secuencia,secuencia2;
float ref_pH=0.0;
float error_pH=0.0;

const float SalidaResistencia[5][5]={ {0 , 0 , 0 , 900 , 900},
                                         {0 , 0 , 0 , 900 , 900},
                                         {0 , 0 , 0 , 900 , 900},
                                         {0 , 0 , 0 , 450 , 900},
                                         {0 , 0 , 0 , 225 , 900}};

//CONSTANTES DETECCION DE CRUCE POR CERO
int16 val_tmr0;

/*****/
#task(rate=12ms, max=10ms)
```

```

Void temp( );

#task(rate=10ms, max=1ms)
Void pH( );

#task(rate=100ms, max=1ms)
Void Reset( );

#task(rate=100ms, max=1ms)
Void Paro( );

#task(rate=100ms, max=1ms)
Void Reposicion( );

/*****
*****/

#int_COMP
void COMP_isr()
{
    IF(C1OUT){
        set_timer0(val_tmr0);
    }else if(!C1OUT){
        set_timer0(val_tmr0);
    }
}

#int_TIMER0
void RTCC_isr()
{
    output_high(PIN_E0);
    delay_us(2);
    output_low(PIN_E0);
}

/*****
*****METODOS*****
*****/

void lcd_cursor_on()
{
    lcd.rs=0;
    lcd_send_nibble(0);
    lcd_send_nibble(0x0F);
}

char kbd_get_key()
{
    char key='\0';
    while(key=='\0')
        key=kbd_getc();
    return key;
}

void Bienvenida()
{
    printf(lcd_putc, "Bienvenido");
    delay_ms(500);
    lcd_putc('.');
    delay_ms(500);
    lcd_putc('.');
    delay_ms(500);
    lcd_putc('.');
    delay_ms(500);
    lcd_cursor_on();
}

```

```

}

int get_ref_Temp()
{
    int datos[3];
    int i,key;
    int ref_temp;
    printf(lcd_putc,"\fIngrese el valor\nde T(°C):__\b\b");
    for(i=0;i<=2;){
        key=kbd_get_key();
        datos[2]=key-0x30;
        if(key!='*'&&key!='#'){
            if(i==0||i==1){
                lcd_putc(key);
                datos[i]=key-0x30;
                i++;
            }
        }else if(key=='*'){
            switch(i){
                case 0: break;
                case 1: printf(lcd_putc, "\b\b");
                        i--;
                        break;
                case 2: printf(lcd_putc, "\b\b");
                        i--;
                        break;
            }
        }else if(key=='#'){
            switch(i){
                case 0: break;
                case 1: printf(lcd_putc, "0");
                        datos[1]=0;
                        i=2;
                        break;
                case 2: lcd_putc('\f');
                        i=3;
                        break;
            }
        }
    }
    return ref_Temp=datos[0]*10+datos[1];
}

float get_ref_pH()
{
    int datos[4];
    int i,key;
    float ref_pH;
    for(i=0;i<=3;){
        key=kbd_get_key();
        datos[3]=key-0x30;
        if(key!='*'&&key!='#'){
            if(i==0&&datos[3]<=1){
                lcd_putc(key);
                datos[i]=key-0x30;
                i++;
            }else if(i==1&&datos[3]<=4){
                lcd_putc(key);
                datos[i]=key-0x30;
                i++;
                lcd_putc('.');
            }else if(i==1&&datos[0]<1){
                lcd_putc(key);
                datos[i]=key-0x30;
            }
        }
    }
}

```

```

        i++;
        lcd_putc('.');
    }else if(i==2&&datos[0]==1&&datos[1]==4&&datos[3]==0){
        lcd_putc(key);
        datos[i]=key-0x30;
        i++;
    }else if(i==2&&datos[0]<1){
        lcd_putc(key);
        datos[i]=key-0x30;
        i++;
    }
}
}else if(key=='*'){
    switch(i){
        case 0: break;
        case 1: printf(lcd_putc, "\b\b");
                i--;
                break;
        case 2: printf(lcd_putc, "\b\b_.\b\b");
                i--;
                break;
        case 3: printf(lcd_putc, "\b\b");
                i--;
                break;
    }
}
}else if(key=='#'){
    switch(i){
        case 0: break;
        case 1: printf(lcd_putc, "0.0");
                datos[1]=0;
                datos[2]=0;
                i=3;
                break;
        case 2: printf(lcd_putc, "0");
                datos[2]=0;
                i=3;
                break;
        case 3: lcd_putc('\f');
                i=4;
                break;
    }
}
}
}return ref_pH=datos[0]*10+datos[1]+datos[2]/10.0;
}

void CalcularPertenencia(float error,float dvariable,)
{
    int i;
    const float GraficaDTemp[5][3]={ { -1 , -1 , -0.5 },
                                        { -1 , -0.5 , 0 },
                                        { -0.5 , 0 , 0.5 },
                                        { 0 , 0.5 , 1 },
                                        { 0.5 , 1 , 1 } };

    const float GraficaError[5][3]={ { -100 , -100 , -25 },
                                       { -50 , -25 , 0 },
                                       { -10 , 0 , 10 },
                                       { 0 , 20 , 50 },
                                       { 25 , 100 , 100 } };

    for(i=0;i<5;i++){
        if (i==0 && error<=GraficaError[i][2]||i==4 &&
error>=GraficaError[i][2]){
            PertenenciaError_temp[i]=1;

```



```

    }
    else{
        if(error<GraficaError[i][2]&&error>GraficaError[i][0]){
            if(error<=GraficaError[i][1]){
                PertenenciaError_temp[i]=(error-
GraficaError[i][0])/(GraficaError[i][1]-GraficaError[i][0]);
            }else{
                PertenenciaError_temp[i]=(GraficaError[i][2]-
error)/(GraficaError[i][2]-GraficaError[i][1]);
            }
        }else{
            PertenenciaError_temp[i]=0;
        }
    }
}

for(i=0;i<5;i++){
    if (i==0      &&      dvariable<=GraficaDTemp[i][1]||i==4      &&
dvariable>=GraficaDTemp[i][2]){
        PertenenciaDTemp[i]=1;
    }
    else{
        if(dvariable<GraficaDTemp[i][2]&&dvariable>GraficaDTemp[i][0]){
            if(dvariable<=GraficaError[i][1]){
                PertenenciaDTemp[i]=(dvariable-
GraficaDTemp[i][0])/(GraficaDTemp[i][1]-GraficaDTemp[i][0]);
            }else{
                PertenenciaDTemp[i]=(GraficaDTemp[i][2]-
dvariable)/(GraficaDTemp[i][2]-GraficaDTemp[i][1]);
            }
        }else{
            PertenenciaDTemp[i]=0;
        }
    }
}

float defuzzResistencia()
{
    int i,j;
    float num_sal=0;
    float den_sal=0;
    float pertenencia_regla = 0;

    for(i=0;i<5;i++){
        for(j=0;j<5;j++){
            pertenencia_regla      =      PertenenciaDTemp[i]      *
PertenenciaError_temp[j];
            if(pertenencia_regla!=0){
                num_sal=num_sal+pertenencia_regla*SalidaResistencia[i][j];
                den_sal=den_sal+pertenencia_regla;
            }
        }
    }return num_sal / den_sal;
}

void Angulo_Disparo(int16 voltaje)
{
    switch(voltaje){
        case 0: val_tmr0=65536;break;
        case 1: val_tmr0=65155;break;
        case 2: val_tmr0=64770;break;
        case 3: val_tmr0=64384;break;
    }
}

```

```
case 4: val_tmr0=64014;break;
case 5: val_tmr0=63636;break;
case 6: val_tmr0=63257;break;
case 7: val_tmr0=62882;break;
case 8: val_tmr0=62489;break;
case 9: val_tmr0=62108;break;
case 10: val_tmr0=61742;break;
case 11: val_tmr0=61365;break;
case 12: val_tmr0=60978;break;
case 13: val_tmr0=60611;break;
case 14: val_tmr0=60205;break;
case 15: val_tmr0=59855;break;
case 16: val_tmr0=59466;break;
case 17: val_tmr0=59070;break;
case 18: val_tmr0=58701;break;
case 19: val_tmr0=58327;break;
case 20: val_tmr0=57948;break;
case 21: val_tmr0=57564;break;
case 22: val_tmr0=57176;break;
case 23: val_tmr0=56819;break;
case 24: val_tmr0=56423;break;
case 25: val_tmr0=56060;break;
case 26: val_tmr0=55657;break;
case 27: val_tmr0=55287;break;
case 28: val_tmr0=54915;break;
case 29: val_tmr0=54541;break;
case 30: val_tmr0=54164;break;
case 31: val_tmr0=53785;break;
case 32: val_tmr0=53404;break;
case 33: val_tmr0=53021;break;
case 34: val_tmr0=52636;break;
case 35: val_tmr0=52250;break;
case 36: val_tmr0=51902;break;
case 37: val_tmr0=51513;break;
case 38: val_tmr0=51124;break;
case 39: val_tmr0=50733;break;
case 40: val_tmr0=50381;break;
case 41: val_tmr0=49989;break;
case 42: val_tmr0=49596;break;
case 43: val_tmr0=49243;break;
case 44: val_tmr0=48850;break;
case 45: val_tmr0=48496;break;
case 46: val_tmr0=48103;break;
case 47: val_tmr0=47709;break;
case 48: val_tmr0=47356;break;
case 49: val_tmr0=46963;break;
case 50: val_tmr0=46571;break;
case 51: val_tmr0=46219;break;
case 52: val_tmr0=45828;break;
case 53: val_tmr0=45438;break;
case 54: val_tmr0=45088;break;
case 55: val_tmr0=44701;break;
case 56: val_tmr0=44314;break;
case 57: val_tmr0=43929;break;
case 58: val_tmr0=43545;break;
case 59: val_tmr0=43163;break;
case 60: val_tmr0=42782;break;
case 61: val_tmr0=42404;break;
case 62: val_tmr0=42027;break;
case 63: val_tmr0=41652;break;
case 64: val_tmr0=41280;break;
case 65: val_tmr0=40910;break;
case 66: val_tmr0=40542;break;
```

```

case 67: val_tmr0=40140;break;
case 68: val_tmr0=39777;break;
case 69: val_tmr0=39382;break;
case 70: val_tmr0=39025;break;
case 71: val_tmr0=38636;break;
case 72: val_tmr0=38250;break;
case 73: val_tmr0=37868;break;
case 74: val_tmr0=37490;break;
case 75: val_tmr0=37117;break;
case 76: val_tmr0=36747;break;
case 77: val_tmr0=36381;break;
case 78: val_tmr0=35987;break;
case 79: val_tmr0=35598;break;
case 80: val_tmr0=35215;break;
case 81: val_tmr0=34837;break;
case 82: val_tmr0=34465;break;
case 83: val_tmr0=34099;break;
case 84: val_tmr0=33708;break;
case 85: val_tmr0=33325;break;
case 86: val_tmr0=32949;break;
case 87: val_tmr0=32580;break;
case 88: val_tmr0=32192;break;
case 89: val_tmr0=31812;break;
case 90: val_tmr0=31440;break;
case 91: val_tmr0=31078;break;
case 92: val_tmr0=30700;break;
case 93: val_tmr0=30309;break;
case 94: val_tmr0=29929;break;
case 95: val_tmr0=29562;break;
case 96: val_tmr0=29185;break;
case 97: val_tmr0=28802;break;
case 98: val_tmr0=28414;break;
case 99: val_tmr0=28044;break;
case 100: val_tmr0=27674;break;
case 101: val_tmr0=27289;break;
case 102: val_tmr0=26912;break;
case 103: val_tmr0=26532;break;
case 104: val_tmr0=26141;break;
case 105: val_tmr0=25775;break;
case 106: val_tmr0=25395;break;
case 107: val_tmr0=25021;break;
case 108: val_tmr0=24642;break;
case 109: val_tmr0=24262;break;
case 110: val_tmr0=23886;break;
}
}

```

```

Void BombaAcido()
{ switch (secuencia){
    case 1: portc=0b10100000;
            secuencia++;
            break;
    case 2: portc=0b10000000;
            secuencia++;
            break;
    case 3: portc=0b10010000;
            secuencia++;
            break;
    case 4: portc=0b00010000;
            secuencia++;
            break;
    case 5: portc=0b01010000;
            secuencia++;

```

```

        break;
    case 6: portc=0b01000000;
        secuencia++;
        break;
    case 7: portc=0b01100000;
        secuencia++;
        break;
    case 8: portc=0b00100000;
        secuencia=1;
        break;
    }
}

Void BombaBase()
{
    switch (secuencia2){
        case 1: portc=0b00001010;
            secuencia2++;
            break;
        case 2: portc=0b00001000;
            secuencia2++;
            break;
        case 3: portc=0b00001001;
            secuencia2++;
            break;
        case 4: portc=0b00000001;
            secuencia2++;
            break;
        case 5: portc=0b00000101;
            secuencia2++;
            break;
        case 6: portc=0b00000100;
            secuencia2++;
            break;
        case 7: portc=0b00000110;
            secuencia2++;
            break;
        case 8: portc=0b00000010;
            secuencia2=1;
            break;
    }
}

/*****TAREAS*****/

void Reset()
{
    if(input_state(PIN_A6)==0){
        rtos_terminate();
    }
    rtos_yield();
}

void Paro()
{
    if(input_state(PIN_A6)==1){
        rtos_disable(temp);
        set_pwm1_duty(0);
    }
    rtos_yield();
}

void Reposicion()
{

```

```

        if(input_state(PIN_A4)==1){
            rtos_enable(temp);
        }
        rtos_yield();
    }

void temp ( )
{
    set_adc_channel(1);
    valor = read_adc ();
    valor *= 500.0 / 1024.0;
    error = k * (ref - valor);
    dtemp = kd * (valor - valor_1);
    CalcularPertenencia(error,dtemp);
    x = defuzzResistencia();
    x *= ko;
    control = x;
    if (control > 110) control = 110;
    Angulo_Disparo(control);
    valor_1 = valor;
    if(ref-valor < -10) output_high(PIN_E1);
    else output_low(PIN_E1);
    rtos_yield();
}

void pH ( )
{
    int valor;
    set_adc_channel(4);
    valor = read_adc ();
    valor = 1 + (valor*4/1024.0);
    error_pH = ref - valor;
    if(error_pH>0.2)
        BombaAcido();
    else if(error_pH<ref-0.2)
        BombaBase();
    else
        PORTC=0;
}

/*****MAIN*****/

void main()
{
    TRISA = 0b11111111;
    TRISB = 0;
    TRISC = 0;
    TRISD = 0;
    TRISE = 0b0111;
    PORTB = 0;
    PORTC = 0;
    PORTD = 0;
    CMCON = 0b00000100;

    setup_adc_ports(AN0_TO_AN4|VREF_VDD);
    setup_adc(ADC_CLOCK_INTERNAL|ADC_TAD_MUL_20);
    enable_interrupts(INT_COMP);
    enable_interrupts(INT_TIMER0);

    secuencia=1;
    secuencia2=1;
    lcd_init();

```

```

Bienvenida();
kbd_init();
while(true){
    printf(lcd_putc, "\fIngrese el valor\nde pH: __. __\b\b\b\b");
    ref_pH=get_ref_pH();
    printf(lcd_putc, "\fIngrese el valor\nde T(°C): __\b\b");
    ref=get_ref_Temp();
    printf(lcd_putc, "pH: %2.1f", ref_pH);
    printf(lcd_putc, "\nTemp: %2u", ref);
    enable_interrupts(GLOBAL);
    rtos_run();
    disable_interrupts(GLOBAL);
}
}

```